A Hands-On Report: Commodore's New 128 Computer

COMPUTE!

$2.95
June
1985
Issue 61
Vol. 7, No. 6
$3.50 Canada
02193
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

Apple SpeedScript
A Powerful Word Processor
Complete And Ready-To-Run Program Inside

Webster Dines Out—
Action Game For Youngsters
Programs For Apple, TI, Commodore 64, VIC-20, Atari, IBM PC, PCjr

How To Buy The Right Printer
Which Features Do You Really Need?

Commodore 64 Disk Editor
Examine And Change Any Disk

IBM Variable Lister
A Valuable Programmer's Tool

Page Flipping On The Atari
Create Exciting Animation

Apple Universal INPUT
Making BASIC Better

0 71486 02193 06

BLAZING NEW

The OKIMATE COLOR

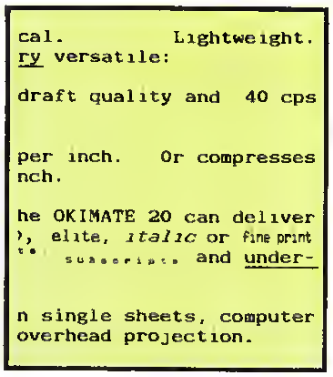
The first affordable color printers!

The new OKIMATE Personal Color Printers are breaking through in flying colors. They're the first low cost personal printers that let you print in rainbows of dazzling colors.

Now your computer can take on new meaning. Because the OKIMATE Printers can bring the information on your screen to life. In brilliant colors. And for very little green.

Fully equipped for reading, writing and 'rithmetic.

The OKIMATE word processing capability delivers crisp, clean business letters, term papers, financial reports and homework. So now you can print in minutes, instead of typing it in hours. You can even highlight words, addresses, paragraphs and charts. Even underline points you want to emphasize. So you and your information really stand out.



Easy to learn. Easier to use.

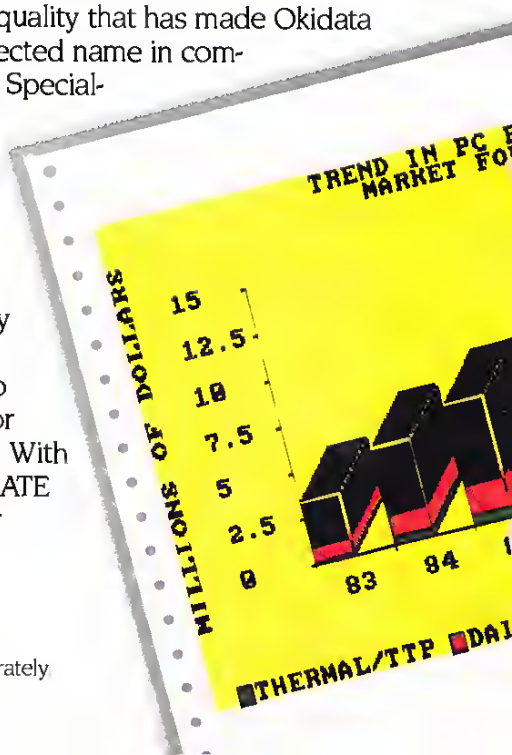
"Learn-to-Print" software packages come with your OKIMATE Printers to show you how to start printing. In fact, the OKIMATES come with everything you need for color printing. Including a data cable, interface board, color ribbon, black ribbon and "color screen print" program on diskette. Now you're set.

Just plug your new OKIMATE Printer into your computer with the PLUG 'N PRINT package*. And print. It's that easy. In minutes you'll be printing everything from financial reports to souffle recipes. Home budgets to original drawings. In rainbows of brilliant colors.

Built and backed by the reliability leader.

The new OKIMATE Personal Printers are the latest example of Okidata's technological craftsmanship. Built with the same dedication to quality that has made Okidata the most respected name in computer printers. Specially designed to be small and lightweight. Operating as quiet as a whisper. And beautifully affordable.

So grab onto OKIMATE color printing today. With the new OKIMATE Personal Color Printers.



*Special PLUG 'N PRINT packages available separately

COLOR IS HERE!

PRINTERS have arrived.

The OKIMATE 10

OKIMATE offers you a colorful choice.

The OKIMATE 10.
Color that brings your computer to life.

The OKIMATE 10 Personal Color Printer prints in over 36 dazzling colors. It's completely compatible with your Atari® and Commodore® personal computers. Comes with a 9 element printhead. And prints a speedy 240 words per minute. For crisp, clean letters, reports, and homework. All this and beautiful color for about $200. Available wherever Atari and Commodore computers are sold.

The OKIMATE 20

The OKIMATE 20.
The color printer that's all business.

The OKIMATE 20 Personal Color Printer is here to dazzle everyone. With the vibrant impact of over 100 sizzling colors. A 24 element printhead that delivers letter quality characters. And the ability to print 270 words a minute for reports, financial statements and letters. It's completely compatible with IBM® PC and PCjr. And for all you Apple owners, the OKIMATE 20 works wonders with the IIE,® the IIC® and the Macintosh.® It's affordable color printing for under $268. Available at computer dealers everywhere.

OKIDATA
an OKI AMERICA company
Mt. Laurel, NJ 08054

Gone are the glory days for Apple. Because Europe's most successful business computer company is now doing business in America.

Introducing Apricot. A full line of computers specifically designed for business.

Not adapted to it.

In fact, the facts speak for themselves.

Apricots are elegant and compact, true 16-bit computers. They employ the MS-DOS operating system, and a minimum of 256K memory. One of our models, the Apricot Xi, boasts an incredible 1 Megabyte of memory, and features a Winchester hard disk with 20 Megabytes of storage. We also have models that feature speech recognition, full-size LCD, and icon driven menus.

In addition, you have a choice between 9″ or 12″ b/w or 10″ color monitors. All of

Present and Future.

which have a higher screen resolution than Apple.

And as if all that weren't enough, all of our models can be networked from the moment you take them out of the box. They're also capable of running thousands of business software programs like Lotus,™ pfs,® and d-Base III.™ Specially written for Apricot on 3½ inch disks.

Now, how do you like them Apricots?

Apricot, Inc., 3375 Scott Blvd., Santa Clara, CA 95054. Call 800-227-6703, or in California 800-632-7979.

The Apricot Portable. 512K RAM. 720K diskette. 80x25 line LCD. MS-DOS. $2495.

apricot™

We're changing how American business does business.

AMAZING DAISY

NOW! FULL SIZE, FULL FEATURE, LETTER QUALITY AT ONLY $353

If you have been searching for a letter quality printer you have probably found the flood of claims and counterclaims to be a real roadblock in your search. Not long ago we were in the same position. We tried to determine which daisy wheel printer had all the features our customers wanted, yet would not set them back a month's salary. Recently several manufacturers have introduced machines that had features we were seaching for. After a thorough assessment, we eliminated one model after the other for lack of one feature or another until we only had one left.

THE RESULTS ARE IN

We found the printer which has all the features anyone could want. The winner is the Aprotek Daisy 1120, a real heavy-duty workhorse printing at 20 characters per second. The manufacturer is Olympic Co. Ltd., a highly respected Japanese firm.

FEATURES GALORE

This printer has it all. To start with, it has a front panel Pitch Selector button with indicators which allows 10, 12, 15 characters per inch (CPI) or Proportional Spacing. There is a Select (Online) button (with indicator) and a Line Feed button. You can also set Top-of-Form or Form Feed with the touch of the TOF button. Other front panel indicators include Power and Alarm.

To load a sheet of paper, simply place it in the feed slot and pull the paper bail lever. PRESTO! The paper feeds automatically to a 1 inch top margin and the carriage aligns to the selected left margin. In this manner, each page can have identical margins automatically. You can continue to compute while the Daisy 1120 is printing. The built in 2K buffer frees up your computer while printing a page or two allowing you to go to your next job.

To really put your printer to work, the Cut Sheet Feeder option is great for automatic printing of those long jobs. Also available is the adjustable Tractor Feed option. Compare our option prices! Best of all the Daisy 1120 is quiet: only 57 dB-A (compare with an average of 62-65 dB-A for others).

COMPLETE COMPATIBILITY

The Daisy 1120 uses industry standard Diablo® compatible printwheels. Scores of typeface styles are available at most computer or stationary stores. You can pop in a 10, 12, 15 pitch or proportional printwheel and use paper as wide as 14". At 15 CPI you can print 165 columns—great for spreadsheets.

The Daisy 1120 uses the Diablo Hytype II® standard ribbon cartridges. Again universally available.

Not only is the hardware completely compatible, the control codes recognized by the Daisy 1120 are Diablo 630® compatible (industry standard). You can take advantage of all the great features of word processing packages like Wordstar®, pfs: Write®, Microsoft Word® and most others which allow you to automatically use superscripts, subscripts, automatic underlining, boldface (shadow printing) and doublestrike.

The printer has a set of rear switches which allow the use of standard ASCII as well as foreign character printwheels. Page length can be set to 8, 11, 12, or 15". The Daisy 1120 can also be switched to add automatic line feed if required.

THE BEST PART

When shopping for a daisy wheel printer with all these features (if you could find one), you could expect to pay $600 or $700 dollars. The options would add much more. *Not now!* We have done our homework. We can now offer this printer for only $353. Order yours today!

NO RISK OFFER

Try the Daisy 1120 for 2 weeks. If you are not satisfied for ANY reason we will refund the full price—promptly. A full 1-year parts and labor warranty is included

THE BOTTOM LINE

Aprotek Daisy 1120 (Order#1120) $353 w/standard Centronics parallel interface and 2K buffer.

Options

Auto Cut Sheet Feeder (#1110) $188

Tractor Feed (#1112) $77

Accessories

8' Cable for IBM PC® and compatibles (#1103) $26

Interface with cable: •TI-99/4A (#106) $66
•Apple II or IIe (#1104) $76
•All Commodore (except Pet) (#1105) $44
•All Atari (#1107) $66

Shipping is $11—UPS continental USA. If you are in a hurry, UPS Blue or Air Parcel Post (second day air) is $25. Canada, Alaska, Mexico and Hawaii are $30 (air). Other foreign is $60 (air). California residents add 6% tax. Prices are cash prices—VISA and M/C add 3% to total. We ship promptly on money orders, cashier's checks, and charge cards. Allow 14-day clearing for checks. No C.O.D.'s. Payment in US dollars only

TO ORDER ONLY CALL TOLL FREE

(800) 962-5800 USA
(800) 962-3800 CALIF. (8-8 PST)

Or send payment to address below:

Technical Information & Customer Service: (805) 987-2454 (8-5 PST)

Dealer Inquiries Invited

APROTEK

1071-A Avenida Acaso, Camarillo, CA 93010

June 1985 Vol. 7, No. 6

FEATURES

			GUIDE TO ARTICLES AND PROGRAMS
18	The Commodore 128: A Hands-On Report	Tom R. Halfhill	•
30	How to Buy the Right Printer	Kathy Yakal	•
36	Solving Common Printer Problems	Selby Bateman	•
42	Webster Dines Out	Walter Bulawa	64/VIC/AT/PC PCjr/AP/TI
116	*SpeedScript 3.0*: All Machine Language Word Processor for Apple	Charles Brannon and Kevin Martin	AP

REVIEWS

58	*The Hitchhiker's Guide to the Goloxy*	Neil Randall	64/AT/AP/Mac/PC PCjr/TI/Kaypro
58	*Super-Text*	Arthur Leyenberger	64/AT/AP/PC
65	*War in Russia*	Neil Randall	AT/AP
66	*Raid on Bungeling Boy*	James V. Trunzo	64
66	*Sundog: Frozen Legacy*	James V. Trunzo	AP
67	*Enhancements to BASIC* for Atari	Tom R. Halfhill	AT

COLUMNS AND DEPARTMENTS

6	The Editor's Notes	Richard Mansfield	•
10	Readers' Feedback	The Editors and Readers of COMPUTE!	•
68	The Beginner's Page	Tom R. Halfhill	•
70	Computers and Society: Expert Systems and the Mass Market Micro	David D. Thornburg	•
72	On the Road with Fred D'Ignazio: Bits, Bytes, and Black Sheep	Fred D'Ignazio	•
73	Telecomputing Today: Inside XMODEM	Arlan R. Levitan	•
74	IBM Personal Computing: Escaping on a LaserJet	Donald B. Trivette	PC/PCjr
75	INSIGHT: Atari—Analyzing the BASIC Bug	Bill Wilkinson	AT
76	Programming the TI Multiple Choice Test	C Regena	TI

THE JOURNAL

78	Housepainter: Inverted Video on the Commodore 64	Jim Butterfield	64
82	BASIC File Editor for Commodore	Henry A. Doenlen	64/VIC
84	Page Flipping on the Atari	Clay Stuart	AT
87	Commodore 64 Hi-Res Quick Clear	Paul W. Downing	64
88	Unlocking IBM BASIC Programs	Peter F. Nicholson	PC/PCjr
89	Fast Atari Circles	Owen Sexsmith	AT
91	Apple Universal INPUT	William Simpson	AP
92	Hardcopy Sprites for Commodore 64	Thomas H. White	64
93	IBM Variable Lister	Peter F. Nicholson	PC/PCjr
96	Apple IIc RAM Disk Mover, Part 2	Christopher J. Flynn	AP
98	Commodore Disk Editor	Martin Sikes	64
102	TI SuperFont	Patrick Parrish	TI
106	Apple ProDOS Variable Lister	Paul F. Stuever	AP
108	Atari Cassette Filenames	Norman Lin	AT

110 COMPUTE!'s Guide to Typing In Programs
114 Apple MLX: Machine Language Entry Program
127 CAPUTE! Modifications or Corrections to Previous Articles
128 Advertisers Index

NOTE: See page 110 before typing in programs.

AP Apple, **Mac** Macintosh, **AT** Atari, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **CC** Radio Shack Color Computer
*General interest

TOLL FREE Subscription Order Line
800-334-0868 (in NC 919-275-9809)

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019
Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 914, Farmingdale, NY 11737. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X.

EDITOR'S NOTES

With this issue, COMPUTE! has a new look. Several adjustments to the way we put together the magazine have been made in the past few months, and this issue implements a process which started—on the drawing boards—last year.

The personal computer marketplace is maturing and currently pausing for breath after several frenetic years in the early 1980s. Many manufacturers, and many computer magazines, have retired from the scene. There are approximately one-fifth as many hardware and software companies today as there were a year ago.

For magazine publishers, this greatly diminished universe of advertisers represents a challenge. Fewer ads necessarily mean fewer pages.

While there are dozens of popular magazines like *High Fidelity* and *Science Digest* which have stabilized over the years at fewer pages than COMPUTE! currently prints, there are some economies which we must now effect. In a nutshell, we want to continue to bring our readers an equivalent amount of information in a smaller package every month.

Fortunately, there are several ways to seek painless concision. The first thing we looked at was the size of our typeface, the "point size." COMPUTE! has always printed larger characters than is traditional for magazines of its class and audience. It may not be too easy to detect, but with this issue we have slightly reduced the type by one point in most of the magazine (one point equals 1/72 inch). "Reviews," "Readers' Feedback," and the "News & Products" sections have been reduced by two points. These are gentle reductions and bring COMPUTE!'s type in line with most other popular magazines.

However, even though there will be no decrease in readability, this change results in an average of 10 to 15 percent more information per page.

To maintain the ease with which COMPUTE! can be read, we have made additional changes to our layout. For one thing, we have gone to a primarily three-column format, replacing our previous two-column design. Program listings, too, have changed from two- to three-column format. Aside from contributing more words per page while still maintaining easy readability, these adjustments give our design staff greater flexibility to create layouts with more variety and eye appeal.

Finally, we have requested that our columnists write within one- or two-page limits each month. With the more economical type size and page layout, they will be able to deliver excellent information each month in less space. Ultimately, that frees the editors to increase the variety and content of COMPUTE!.

In a not entirely unrelated development, Philip Nelson has accepted the position of assistant editor of COMPUTE!. Philip has been on our staff as assistant technical editor for six months and has proven to be an excellent, careful editor and a fine writer as well.

We feel that these changes to COMPUTE! are both necessary and positive. These adjustments will allow us to continue to bring you the greatest number of high quality articles and programs in the years to come.

Senior Editor

Educational Software
That Works:

Spell.

Spell It!

Spell expertly 1000 of the most misspelled words. Learn the spelling rules. Improve with 4 exciting activities, including a captivating arcade game! Add your own spelling words.

ages 10 — adult / 2 disks: $49.95

NOW! FOR ATARI

Math.

Math Blaster!

Master addition, subtraction, multiplication, division, fractions, decimals and percentages — by solving over 600 problems. Learn your math facts with 4 motivating activities, including a fast-action arcade game! Add your own problems.

ages 6 — 12 / 2 disks: $49.95

NOW! FOR ATARI

Word.

Word Attack!

Add 675 new words to your vocabulary — with precise definitions and sentences demonstrating usage. Build your skills with 4 fun-filled activities, including an arcade game! Add your own words.

ages 8 — adult / 2 disks: $49.95

Read.

Speed Reader II

Increase your reading speed and improve comprehension! Six exercises designed by reading specialists vastly improve your reading skills. Chart your own progress with 35 reading selections and comprehension quizzes. Add your own reading materials.

high school, college & adult / 2 disks: $69.95

The Davidson Best Seller Tradition.

For your Apple, IBM or Commodore 64
Ask your dealer today.

For more information call: (800) 556-6141
in California call: (213) 373-9473

Davidson & Associates, Inc.
6069 Groveoak Place #12
Rancho Palos Verdes, CA 90274

Davidson.

teacher certified NEA™

Coming In Future Issues

Apple *SpeedScript 3.0* ProDOS Converter

Softball Statistics For Atari, Commodore 64, VIC-20, Plus/4, 16, PET, Apple, IBM PC, PCjr, TI-99/4A

Chess For IBM PC & PCjr

Commodore 64 Disk Commander

Adding TIME$ To Atari BASIC

Apple Fractals

The Last Warrior Action Game For IBM PC, PCjr, Commodore 64, Atari, Apple

COMPUTE! Publications, Inc. publishes:

COMPUTE!
COMPUTE!'s GAZETTE
COMPUTE! Books
COMPUTE!'s GAZETTE DISK

Corporate office:
324 West Wendover Avenue
Suite 200
Greensboro, NC 27408 USA

Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA

Telephone: 919-275-9809

Subscription Orders

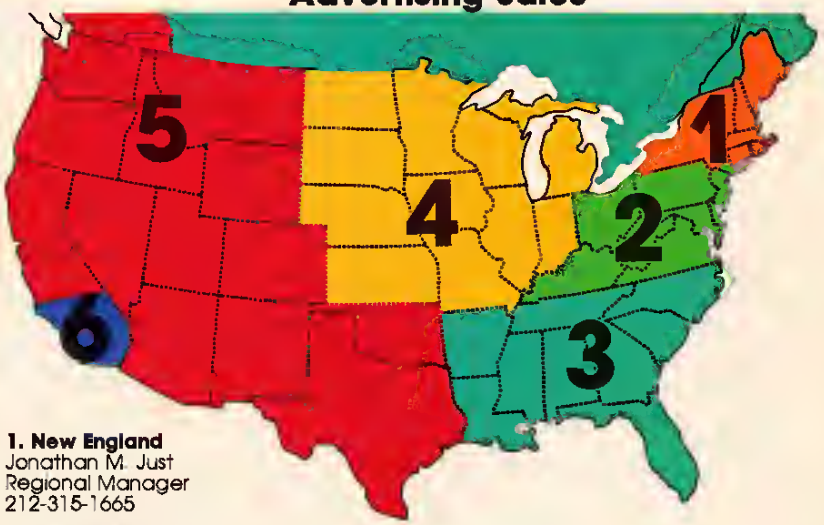
COMPUTE! Circulation Dept.
P.O. Box 914
Farmingdale, NY 11737

TOLL FREE Subscription Order Line
800-334-0868
In NC 919-275-9809

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$24
	(two yrs.)	$45
	(three yrs.)	$65
Canada and Foreign Surface Mail		$30
Foreign Air Delivery		$65

Advertising Sales



1. New England
Jonathan M. Just
Regional Manager
212-315-1665

2. Mid Atlantic
John Soval
Eastern Advertising Manager
212-315-1665
Kathy Hicks
Morsha A. Gittelman
215-646-5700
Brian S. Rogers
212-674-0238

3. Southeast & Foreign
Harry Blair
919-275-9809

4. Midwest
Gordon Benson
312-362-1821

5. Northwest/ Mountain/Texas
Phoebe Thompson
408-354-5553

6. Southwest
Ed Winchell
213-378-8361

Director of Advertising Sales
Ken Woodard

COMPUTE! Home Office 919-275-9809

Address all advertising materials to:
Patti W. Stokes
Advertising Production Coordinator
COMPUTE! Magazine
324 West Wendover Avenue,
Greensboro, NC 27408

The COMPUTE! subscriber list is made available to carefully screened organizations with a product or service which may be of interest to our readers. If you prefer not to receive such mailings, please send an exact copy of your subscription label to: COMPUTE! P.O. Box 914, Farmingdale, NY 11737. Include a note indicating your preference to receive only your subscription.

Available Now!
Trivia Fever
Volume 2
and
Super
Sports

Catch Trivia Fever™

"The Hottest New Game In Town"

At $39.95, Trivia Fever comes complete with Question and Answer Book, Category Selector, and Tally Sheets to be used when played without a computer.

Trivia Fever is absolutely unique — it's the only software entertainment package that can be enjoyed ***with*** or ***without*** a home computer! When played on your home computer, Trivia Fever is a refreshing alternative to all those shoot'em up games. An elected "Master of the Game" uses the computer to randomly select subject categories, handicap players, generate questions and answers, keep score automatically, and more! Instructive by its very nature, Trivia Fever can be enjoyed by up to 8 individuals or teams. And when played without a computer, Trivia Fever has all the best features of the "popular" trivia games plus more — all without the cumbersome board, cards, and little game pieces. You can play in a car, on vacation, anytime, anywhere! And Trivia Fever is by far the best Trivia game available anywhere. Here's why:

Trivia Fever offers thousands of challenging questions in 7 interesting categories, so there's something for everyone. Each category has questions with 3 levels of difficulty, which score comparable points. What's more, Trivia Fever allows players to HANDICAP all those so-called "trivia experts" three different ways, giving everyone a chance to win. And players can easily control the length of play from quick thirty minute games to multi-hour party marathons!

Trivia Fever is unique, entertaining, educational, and most of all FUN. And at $39.95, Trivia Fever is destined to quickly become the best selling software entertainment package of all time. There's even a $5 rebate available to any non-computer users who return the computer diskette.

Trivia Fever can be enjoyed on the Commodore 64, IBM PC & PCjr and compatibles, Apple II series, and others. So don't delay. Catch Trivia Fever at your favorite software retailer today!

For additional information call 617-444-5224, or write to:

PSI P.O. Box 533
Needham, MA 02194

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

Closing The Quality Gap

I thought *letter quality* was a term used to describe typewriters or daisywheel printers. Now I have seen many dot-matrix printers claiming to be letter quality. How is this possible?

Alex Cutrone

Perhaps you're reading too much into the term letter quality. *It simply means "good enough to use in a business letter." Printing of this quality has traditionally been equated with daisywheel printers, which create crisp, solid images by striking an inked ribbon with little character-shaped hammers, just like a typewriter.*

Dot-matrix printers use a different technique, forming each character out of many tiny dots. The printhead contains several small pins which can be individually fired, pressing the ribbon against the paper to make a dot. As the printhead moves across the paper, the pins are rapidly fired in various combinations to form different characters.

Early dot-matrix printers left noticeable gaps between the character dots, giving the print a grainy appearance. As dot-matrix technology has improved, these gaps have been all but eliminated, producing print of much higher quality.

One way to improve print quality is to double-strike *each character. Printing a character twice in the same spot puts more ink in each dot; since the dots are darker (and a little fatter), the print looks better.* Enhanced *printing also double-strikes each character, but offsets the printhead very slightly (less than a dot's width) before printing the character a second time. This fills in much of the space between dots.*

Most dot-matrix printers have pins that are round in cross section. Since round dots don't fit neatly together, dot-matrix characters tend to have wavy edges, even in enhanced printing modes. To give the characters crisper edges and further alleviate the gap problem, some manufacturers have switched to pins with a squarer cross section. However, you can obtain high quality print from a machine that uses round pins: The Apple Imagewriter is one example.

By looking closely, you can still distinguish the best dot-matrix print from so-called letter-quality print. A letter produced with a Macintosh and an Apple Imagewriter doesn't look exactly the same as one typed on an IBM Selectric. But that's not to say it doesn't look good enough for "serious" use. Some people would say that the Macintosh-generated letter looks more *impressive than one done on a Selectric.*

The fact is that many people are already using dot-matrix printers for business correspondence. Judged by that practical standard, the better dot-matrix printers are indeed letter-quality machines. If you're thinking of purchasing a printer, see "How to Buy the Right Printer" elsewhere in this issue.

Runaway IBM Keys

My new PCjr is great except for one irritating feature. When I rest my finger on a key, the computer prints a long line of identical letters. When I press the backspace key to erase the extra letters, it repeats, too. How can I stop this repeating feature, or at least slow it down?

Peter Gliewell

You can find the answer to this and many other questions about IBM Personal Computers in the new book from COMPUTE! Publications, Mapping the IBM PC and PCjr *by Russ Davies. Most microcomputers supply a repeating function for only a few editing keys such as the cursor controls or space bar. The PCjr has what IBM calls a* typamatic *keyboard: Nearly every key repeats when you hold it down. (The exceptions are keys such as Enter, which you would never want to repeat.)*

Depending on your tastes, the typamatic function is either a godsend or a curse. The following program slows down or eliminates your PCjr's typamatic function:

```
100 PRINT "Enter a number to
    adjust typamatic"
110 PRINT " 0  -  Return to n
    ormal"
120 PRINT " 1  -  Increase in
    itial delay"
130 PRINT " 2  -  Half rate o
    f repeat"
140 PRINT " 3  -  Both 1 and
    2 above"
150 PRINT " 4  -  Typamatic f
    unction off"
160 INPUT X:IF X<0 OR X>4 GOT
    O 160
170 X=X*2:DEF SEG=0
180 Y=(PEEK(&H488) AND (&HFF-
    &HE))
190 POKE &H488,(Y OR X)
200 PRINT "PCjr typamatic fun
    ction now adjusted"
```

The typamatic function uses two different time delays. There is a short initial delay between the time you first press a key and the time it begins to repeat. After repeating begins, there is another slight delay between each repetition. The program lets you adjust either or both of these delays. Note that the typamatic function affects the entire keyboard: If you turn it off, none of the keys—including the space bar and cursor controls—will repeat.

Commodore Disk Patterns

I have just finished reading your article about pattern matching on the Commodore disk drive. In the last paragraph of the section labeled "New Patterns," you talk about loading just the disk title, its ID, and the number of blocks free from the directory. Your suggestion is to use a filename not on the disk, such as 0:#$&!%. I have found an easier way : Instead of the usual command LOAD "$",8, use LOAD "$$",8. On a disk with a large number of programs, you cannot use this twice in a row or you'll get a FILE NOT FOUND ERROR. If you load the directory normally (LOAD"$",8) after using LOAD"$$",8 you may use the latter command again.

George Sherman, Jr.

You've stumbled across a curious aspect of Commodore's disk directory system: a few words of explanation might be in order.

Maxell Gold.

The floppy disk that makes Commodore® more commanding, Apples® crisper and gives IBM® PC™ a higher IQ.

Smart move, buying a computer. To be sure your computer doesn't underachieve, put it on the Gold Standard. Maxell. An industry leader in error-free performance. There's a Maxell Gold Standard floppy disk for virtually every computer made. Each comes backed by a lifetime warranty. Maxell. To keep your computing at the top of the class.

maxell®
IT'S WORTH IT.

PC is a trademark of IBM Corp

Maxell Corporation of America, 60 Oxford Drive, Moonachie, N.J. 07074

First: "$" is a legitimate filename. Try writing a short program and then saving it with SAVE "0:$",8. No problem; the file will save without any errors. You'll find it in the directory, but note that LOAD "$",8 brings in the directory, not the program. But you can still get the program with LOAD "0:$",8.

Second: You may pattern-match without using the drive designator, although this will give you trouble. If you load a directory with LOAD "$P",8, you will get all files starting with the letter P. It's much better to command LOAD "$0:P*",8, which specifies the drive number. The reason for this is odd, and relates to the fact that the 1541 DOS (Disk Operating System) software was developed from that found in Commodore's older 4040 dual-drive unit. Single-slot Commodore drives don't know that they are single; they suspect that there's a second drive around somewhere (drive 1, which doesn't exist). So if you use type LOAD "$P*",8, the disk drive will correctly get the directory from drive 0, and then try for a directory from nonexistent drive 1! It won't get it, of course, but it's left in "drive 1 mode," and the next job it gets without a drive specification will try drive 1 first.*

Thus, if you enter LOAD "$P",8 again, the unit will go to drive 1, find no directory, and report FILE NOT FOUND. This won't happen if you command LOAD "$0:P*",8 since drive 0 is forced.*

Let's put these two together. When you type LOAD "$$",8 you are really telling the computer to look for a file with a name of "$". Normally, it won't exist, so you'll get only the disk header and block count; but if you saved a file with that name as suggested above, you'll see it in the directory.

Illogical Apple Logic?

In the August 1984 issue of COMPUTE!, the ANDing and ORing of numbers was explained in the Readers' Feedback section. I tried doing some of the examples on my Apple IIe but I never got the result I expected. For instance, I POKEd the number 15 into location 7, then typed POKE 7,PEEK(7) OR 240. When I checked location 7 by typing PRINT PEEK(7), the result was 1. This also happened when I tried to AND a number. Why is this happening?

Michael Kurtz

On most computers, the BASIC statement PRINT 240 OR 15 gives the result 255. However, Apple (and Atari) computers have a different way of doing the logical operations AND and OR.

Instead of doing a separate AND or OR for each bit of the two numbers, Apples do only one logical operation, treating each of the numbers as a single logical value. A number whose value is zero is considered logically false. *Any other value is treated as* true.

The result of an OR operation is 0 only if both of the numbers are 0. An AND operation gives a result of 1, meaning true, if neither of the numbers is 0. Apple's representation for true is also unusual. Most computers use −1 to represent a true value, but Apple has chosen to use 1.

None of this is important in the situation where logical operators are used the most, conditional statements such as:

```
10 IF A<B AND A<C THEN PRINT "A
   IS BIGGEST"
```

Here, the Apple behaves the same as any other computer. The difference matters only if you want to operate on individual bits of a variable or memory location.

Atari Tape Tribulation

I recently got an *Assembler Editor* cartridge for my Atari 800. I tried out the example programs in the manual and followed Atari's instructions on entering source code and assembling to cassette using ASM,,#C:. After replacing the *Assembler Editor* cartridge with BASIC, I was not able to retrieve the machine language program with CLOAD as specified. I tried several things such as ENTER#C:, but nothing worked. When I did PRINT PEEK(1536), I always got 0. Any suggestions? My temporary alternative is to convert the hexadecimal object code from the ML program into decimal DATA statements using a BASIC program.

Doug Wilson

The manual is incorrect. You cannot use CLOAD, ENTER, or any simple Atari BASIC command to read machine language from tape. Your alternative is actually the best way to include short machine language routines in a BASIC program. However, there is a way to read machine language object files produced by the Assembler Editor. *Unfortunately, when assembling to cassette the tape keeps going during the first pass of the assembly, creating an excessively long leader tone. This makes the tape quite difficult to read.*

Instead of assembling directly to tape, assemble to memory and use the command SAVE #C:<start,end (where start and end correspond to the hexadecimal starting and ending addresses of your program). This will save the object code from memory to cassette. In BASIC, you can use this program to read and POKE the tape program into memory:

```
100 OPEN #1,4,0,"C:"
110 GET #1,A:GET #1,A
120 TRAP 140:GET #1,A:GET
    #1,B:BEG=A+256*B:GET
    #1,A:GET #1,B:FIN=A+
    256*B
130 FOR I=BEG TO FIN:GET
    #1,A:POKE I,A:NEXT I:
    GOTO 120
140 CLOSE #1:TRAP 40000
```

This program will also load a disk object file if you change the device specification in line 100 and add the appropriate filename. Beware of loading ML programs that would overwrite the BASIC program.

1540 Disk Drives With A 64

I had a VIC-20 with a 1540 disk drive. Recently I decided to buy the Commodore 64. This leaves me with a big problem, which I hope you can help me with. Will the 1540 work with the 64? So far I haven't been able to get it to work.

Dave Lester

The best long-term solution is to see if your dealer can get you an upgrade ROM chip that will convert your 1540 to a 1541.

In the meantime, you should be able to save to disk without trouble. To load, you can use a trick to establish communications:

Instead of typing **LOAD "filename",8** *to bring a program into the 64, type* **POKE 53265,43: LOAD "filename",8**. *The screen will go blank, but the file should load correctly. When everything stops, type* **POKE 53265,27** *(you'll have to do this without being able to see what you're typing) and the screen will return to normal.*

The problem arises because the 64 is not able to transfer data to and from the drive as fast as the VIC-20 can. The 64 is slower because more time is required to maintain its 1000-character video display, as opposed to the VIC's 506-character display. When its screen is blank, the 64 operates at the same speed as the VIC-20, and loading can proceed without timing difficulties.

Garbage Collection And Backups

I have seen several references to "garbage collection" in connection with the operation of the Commodore 64, but I have not seen the same term used in connection with operation of the Apple, Atari, or IBM computers. Is the garbage collection routine peculiar to the Commodore, or does it exist on these machines as well?

I have also seen several programs advertised for the 64 which claim to allow you to copy an entire disk. This command exists in DOS on the IBM and Atari machines. Do these programs have some other purpose? Similarly, some of the software for the 64 says it will allow you to copy files from one disk to another. Does this command also not reside in DOS on the 64?

Pat O'Neil

The mysterious-looking term garbage collection *has to do with strings. Strings are tricky for a computer to handle. The computer must set aside enough room in memory for all the strings declared in a program. The length of a string can change*

GET SERIOUS!

FOR COMMODORE AND THE NEW 80-COLUMN DISPLAY FOR ATARI

Transform your C-64™ into a powerful, serious business computer with these three add-on tools.

BusCard II

Mix and match the hardware peripherals of your preference. Increase your programming power with easier-to-use disk commands and machine language. All with one plug-in module.

- lets you use almost any combination of Commodore-compatible floppy or hard disk drives and Centronics-type parallel printers
- all interface functions and device allocations are set by switches on the BusCard II module; errors due to software incompatibility are eliminated
- includes BASIC 4.0, the same powerful language used in Commodore's top-of-the-line business computers, plus a machine language monitor

B.I.-80 Column Adaptar far PaperClip and The Consultant

Double your screen capacity when using your favourite Batteries Included software programs. B.I.-80 turns your 40-column screen into a crystal-clear, high-visibility 80-column display. Works with PaperClip word-processor and The Consultant database manager—maximum readability and minimum eye-strain, even with a screen full of characters. It's the fast, easy, plug-in way to get twice the omount of data onto your monitor screen. And B.I.-80 also gives you the easier-to-use disk commands, with BASIC 4.0 language built right into the module.

- fully self-initializing, no commands to enter; just plug B.I.-80 into the cartridge slot and you're ready to run
- switch back and forth between 80 and 40 column display at any time
- completely eliminates snow, fuzziness, hashing and interference
- easy-to-install module incorporates highest quality hardware components throughout; one-year manufacturer's warranty is standard
- comes complete with 80-column operating system and comprehensive documentation, including full description of BASIC 4.0 commands
- use with Commodore 1701 and 1702 color monitors, or any monochrome video monitor

NEW! B.I.-80 Column Adaptor coming soon for Atari XL computers!

B.I. printer interface

Use the printer of your choice with your Commodore computer—just plug in this compact module, and you're instantly compatible! Take advantage of today's high-speed, high-quality printers with the B.I. Printer Interface.

- works with any Centronics-type parallel printer, which includes almost all major printers on the market
- completely self-contained and ready to run; no extras to buy, no hardware or software modifications to printer or computer are required
- all print functions are controlled by switches on the module; just set them once, and never worry about it again

USE THESE TOOLS WITH PaperClip, The Consultant, AND OTHER HARD-WORKING, HIGH-QUALITY SOFTWARE PROGRAMS FROM BATTERIES INCLUDED

BATTERIES INCLUDED

"The Energized Software Company!"

WRITE FOR A FULL COLOR BROCHURE

186 Queen St West
Toronto, Ontario,
Canada M5V 1Z1
(416) 596-1405

17875 Sky Park North, Suite P
Irvine, California
USA 92714

How to avoid paying your bills.

by Alan Greenspan

Alan Greenspan, Famous Economic Advisor

"The other day, a prominent politician in the executive branch of our government phoned me up.

'Alan,' he said to me, 'the budget is a mess.'

'No joke,' I said.

'Not that budget,' the prominent politician continued. 'My budget. My checking's overdrawn. They're threatening to disconnect my phones. I even got into a shouting match with my wife when I tried to lay off the servants.'

'Civil?'

'Not very. And I think I'm about to be audited. What would I show them? Who keeps receipts for embassy parties?'

At this point, we were disconnected. And although it was too late to teach proper money management to this prominent politician, there is a lesson all of us can learn from his misfortune.

Everyone has to pay their bills, and nobody likes to do it.

You can keep file folders full of bills, drawers stuffed with grocery receipts, envelopes brimming with cancelled checks, and at the end of the month, it still takes hours to figure out just where your money has gone. Not to mention how long it takes to straighten things out at the end of the year.

Well, after years of financial consulting, I've discovered a way to avoid paying your bills: let an Apple® II Personal Computer pay them for you.

There are several advantages to letting an Apple handle your finances.

It will save you time.

It will organize everything.

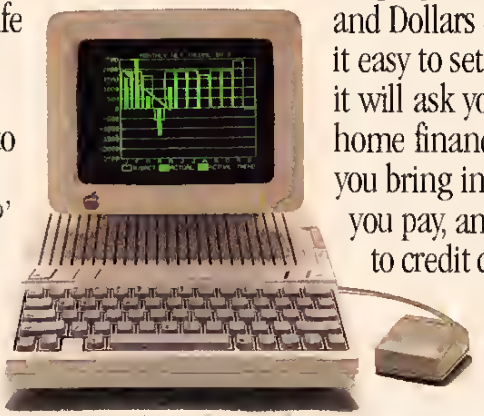
It will tell you, at a glance, exactly what is going on with your money.

It will pay your bills, and never send you any.

And now, I'd like to turn the page over to those nice people at Apple, who will explain, in their own excruciating detail, just what I'm talking about."

The Apple II and the Home Budget.

With software programs like The Home Accountant™ and Dollars & Sense,™ the Apple II makes it easy to set up household books. First, it will ask you some questions about your home finances. Like how much money you bring in each month, how much rent you pay, and whether you owe money to credit card companies, mortgage holders, or any other surly characters. Then, it will ask you to enter some of the bills you receive each month whose prices may vary:



An Apple II will take care of everything from your household budget to your taxes with software programs like Dollars & Sense, The Home Accountant and Tax Preparer

phone, utilities, and the like. Then, it will ask you where you keep your money, and for the numbers of your various checking and savings accounts.

That's really all there is to it. After that, an Apple II can automatically write checks for all your fixed expenses each month. It will also tell you what other bills you can be expecting, and when you enter their costs, an Apple II will pay them, too.

An Apple II will see to it that your checkbooks remain balanced, and that you'll know when your expenses are about to exceed your income. It can even help you plan to buy a new car. Or a home.

Or a fur-lined boat, if your budget permits.

With our Scribe® color/graphics printer you can automatically print out your own checks—not to mention reports, papers, almost anything. Except money.

How to avoid your banker.

After the Apple II writes your checks, it can call your bank with the help of your telephone and an Apple modem. And faster than a teller can say "Next window, please," you can find out all your balances, enter deposits, see what checks have cleared, transfer money from one account to another, and even pay off some of your credit cards and other bills electronically—without ever writing a check.

So the only time you'll have to go to the bank is when you want to visit with your money, personally.

Which, when done in moderation, we can recommend most highly.

*It can manage your entire stock portfolio with programs like Dow Jones Investor's Workshop™ and Charles Schwab and Company's The Equalizer™. It can even show you what's going on in your bank account**

The Apple II and making money.

An Apple II can do wondrous things for your personal finances. With several different software programs, you can become your own stockbroker. Again, by

This is an Apple modem. Not much to look at, we admit, but it does let you pay bills and trade stocks by phone. It also connects your Apple II to a wealth of information services, like THE SOURCE℠ and CompuServe.®

using an Apple modem, you'll gain instant access to financial news sources like *The Wall Street Journal, Barrons,* and the Dow Jones News/Retrieval® service. Find out what they've been saying on *Wall $treet Week.* And in most cases, get up to the minute price quotes on over six thousand stocks, options, and other securities.

An Apple II lets you buy and sell securities right in your home or office, at the moment you want to make the trade. It automatically updates your portfolio and gives you detailed holding reports. It even produces charts and graphs, so you can quickly see how you and your investments are doing.

A little tax relief.

If you become perturbed everytime the subject of doing taxes comes up, an Apple II can do them for you with programs like Forecast™ and Tax Preparer.™

It can store your records, plan for the next year, and calculate your taxes.

You'll be alerted to payments you've made over the year that may be tax-deductible. It even keeps year-round records, automatically updating totals and making corrections for you. It will even print out completed tax forms that the I.R.S. will accept.

And it can do about 10,000 other things totally unrelated to taxes or this ad. So there's no telling how far an Apple II can take you.

"Well, I think that about covers it. And what if, after all of this, you still have some money left over?

Congratulations. You're doing a lot better than the government."

**A note to Dr. Greenspan's relatives. He says, "Don't get excited. This isn't my real bank account."*

as a program runs, causing memory allocation problems. Some BASICs, such as Atari BASIC, tackle this problem by making you DIMension every string before using it for the first time. This sets the maximum length of the string. Atari strings always end up at a single spot in memory after the program starts, and do not move.

Microsoft BASIC uses a different trick. When a string is created, the actual string is stored at the top of BASIC memory. The string's name, length, and a pointer to the address of the string are stored after your program in memory. If you change the string, the new string is copied to the current top of memory (below any other strings) and the pointer is changed. Sooner or later, the strings fill all memory from the top until they collide with variables, arrays, or the end of your program.

This is where garbage collection steps in. A garbage collection routine in BASIC looks at each string, finds the string data, and repacks each string back to the top of memory, discarding unused strings along the way. This can take a while. It's difficult to predict just when garbage collection will occur, and when it does, the machine appears to lock up (in some cases, for more than 20 minutes).

Garbage collection is also a problem on the Apple and IBM machines (and on most Microsoft BASICs), but you can restrict the size of string space on the IBM. This forces garbage collection to occur more frequently, and since the size of the string buffer is small, garbage collection never takes very long. If an IBM program uses a lot of strings, it is sometimes necessary to expand the size of the string area, with the resulting garbage collection problem.

Your question about Disk Operating Systems (DOS) points out some differences between the way a Commodore computer implements DOS and the method used by almost every other computer. Most computers use a RAM-resident DOS to control the disk drive, but Commodore's DOS is in ROM within the drive itself. While this saves user memory, it creates some problems. The 64 and the 1541 drive are like separate computers, and must communicate over a slow serial bus. Commands are sent to the drive to scratch, rename, format, etc. There is a built-in copy command, but it can only copy a file to the same disk. The drive has no way to directly communicate with another disk drive, so the computer must act as a go-between.

The programs you mention let you copy files or disks by reading the disk into computer memory, then copying from memory to another disk. Since Commodore DOS is in the disk drive, there are no built-in programs in the 64 to perform transfers between disks—hence the need for the programs you mention. Many disk duplicators also try to defeat copy-protection schemes, since it's otherwise impossible to back up commercial software.

IBM Hidden Files

I have an IBM PC/XT with one floppy disk and a 10-megabyte hard disk. When I ran CHKDSK on both drives, the computer said I had two hidden files taking up about 22K. What are hidden files, and are they accessible?

Dennis Heckman

When you power up an IBM personal computer, it performs a complex series of housekeeping operations before turning the system over to you. This process usually involves loading two hidden, read-only system files *named IBMBIO.COM and IBMDOS.COM from disk. These are machine language programs which the system needs to operate correctly.*

IBMBIO.COM is an extension of the PC's operating system—called BIOS for Basic Input/Output System—most of which is permanently stored in ROM (Read-Only Memory). The IBMBIO.COM program loads and executes when the system boots up, to initialize input/output devices and perform other system tasks. This is done to correct errors (and there are some) permanently "wired into" the ROM BIOS, and to allow for new devices such as the PCjr's cartridges. IBMBIO.COM is about 3.5K in length.

The PC's DOS (Disk Operating System) is a separate program, roughly 19K long, for manipulating disk data. Since none of DOS is stored in ROM, it's also necessary to load the IBMDOS.COM program from disk and install it in memory when the system boots up. After initializing the system for BIOS, IBMBIO.COM moves IBMDOS.COM into the correct memory area and transfers control to DOS, which in turn does its own initialization before turning the system over to the user.

Files ending in .COM are system files (distinguished from user files which you can alter at will). Since IBMBIO.COM and IBMDOS.COM are critical to normal operation, they're made invisible to user commands such as DIR (disk DIRectory). Out of sight, out of mind—if you don't know a file exists, you'll be less likely to erase it. Both files are further protected by making them read-only: If you can't write to a file, you're less likely to corrupt it by accident.

It is possible to access system files. After each filename in the disk directory is a file attribute byte *which defines the file type. Using the DEBUG program described in your DOS manual, you can read the disk directory, change a file's attribute byte to remove its protective attributes, and write the modified sector back to the disk. However, few users would have any reason to rewrite a system file, and the risk of inadvertent error is enormous. At best, the error might crash the system; at worst, you might lose an entire disk of data. If you want to try modifying a system file, practice on a disk that doesn't contain any important data. You can find a detailed discussion of system files and the boot process in Chapter 1 of COMPUTE!'s* Mapping the IBM PC and PCjr *by Russ Davies.*

Commodore Monitors

What is the difference between a Commodore 1701 and 1702 monitor?

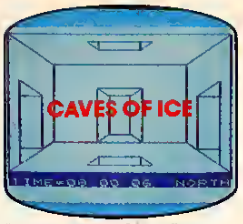
Andy Nagai

There is no appreciable difference between these monitors. The 1701 model changed to 1702 when Commodore began using a different picture tube supplier in late 1983. Cosmetically, it's nearly impossible to tell the two apart. They're virtually identical in appearance, and the electrical connections appear to be the same in both models. We have a number of both models here at COMPUTE!, *and we've noticed that the resolution appears slightly sharper on the 1702s, but this is only because they're newer than the 1701s. (The color on a monitor gradually fades after prolonged use.)*

Commodore also makes the 141 Color Monitor, essentially a 1702 with a charcoal gray color designed to match the Plus/4 and Commodore 16. It's compatible with the VIC-20 and 64. Commodore's newest monitor entries are the 1901 Monochrome Monitor and the 1902 RGBI/Composite Monitor. Each was announced at the Winter CES in support of the Commodore 128.

Get the jump on the weather-man by accurately forecasting the local weather yourself!

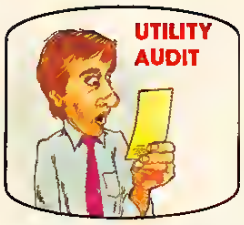
A scientifically proven way to develop an awesome memory.



You are trapped in a five-story, 125-room structure made entirely of ice. Find the exit before you freeze!

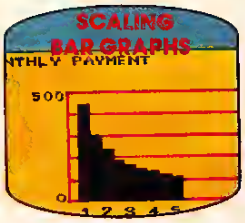
Take control of your personal finances in less than one hour a month.

The beautiful princess is held captive by deadly dragons. Only a knight in shining armor can save her now!



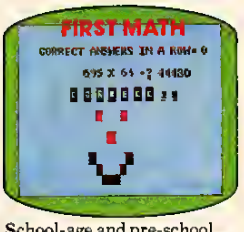
Cut your energy costs by monitoring your phone, electric and gas bills.

Computerize car maintenance to improve auto performance, economy and resale value.

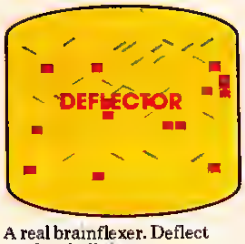


Create multi-colored bar graphs with a surprisingly small amount of memory.

A time-saving organizer for coupons, receipts and more.



School-age and pre-school children are rewarded for right answers, corrected on their wrong ones.



A real brainflexer. Deflect random balls into targets on a constantly changing playfield.

A fun way to dramatically increase typing speed and accuracy.

Get up to 30 new programs and games for less than 15 cents each— every month in COMPUTE!

Every month, COMPUTE! readers enjoy up to 30 brand-new, ready-to-run computer programs, even arcade-quality games.

And when you subscribe to COMPUTE!, you'll get them all for less than 15 cents each!

You'll find programs to help you conserve time, energy and money. Programs like Cash Flow Manager, Retirement Planner, Coupon Filer, Dynamic Bookkeeping.

You'll enjoy games like Air Defense, Boggler, Slalom, and High Speed Mazer.

Your children will find learning fast and fun with First Math, Guess That Animal, and Mystery Spell.

Looking for a challenge? You can write your own games. Customize BASIC programs. Even make beautiful computer music and pictures.

It's all in COMPUTE!. All ready to type in and run on your Atari, Apple, Commodore, PET/CBM, TI-99/4A, Radio Shack Color Computer, IBM PC or IBM PCjr.

What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in joysticks, paddles and track balls?

Order now! Mail the postpaid card attached to this ad and start receiving every issue of COMPUTE!.

For Faster Service Call Toll-Free 1-800-334-0868

COMPUTE! P.O. Box 914, Farmingdale, NY 11737

The Commodore 128: A Hands-On Report

Tom R. Halfhill, Editor

Commodore's new three-in-one machine, the Commodore 128 Personal Computer, should be hitting store shelves in June. In mid-March, COMPUTE! *visited Commodore's U.S. headquarters in West Chester, Pennsylvania to more closely examine the 128, which was announced at the Winter Consumer Electronics Show in January (see "The Next Generation: New Computers at the Winter CES," April 1985). Although a few specifications were not finalized, we found the 128 to be a versatile machine with one of the most powerful BASIC programming languages ever offered in a microcomputer. Here's our report.*

One of the most imitated trends in personal computing lately has been "integrated software"—products that are actually three or more programs in one, like *Lotus 1-2-3*.

Now Commodore is introducing a fresh twist—integrated hardware. With the Commodore 128 Personal Computer, essentially Commodore is wrapping up three computers in one box to sell for under $300 retail. The deceptively small package contains:

1. A standard Commodore 64 with 64K of Random Access Memory (RAM) capable of running virtually all existing 64 software—estimated at 6,000 to 10,000 programs, mostly home and educational.

2. An enhanced Commodore 64 with 128K of RAM and an extremely powerful new BASIC that almost makes PEEK and POKE obsolete.

3. A Z80-based 128K computer designed to run existing software written for the CP/M (Control Program for Microcomputers) operating system—at least 10,000 programs, mostly business/professional.

Expandable to 512K with a RAM disk option, the Commodore 128 also works with all Commodore 64 peripherals as well as a new line of accessories, such as the much faster 1571 disk drive. Rounding out the package are such features as 80-column video in the 128K and CP/M modes, RGB (red-green-blue) high-resolution video output, and all the same ports and interfaces found on the Commodore 64.

Before the Commodore 128 was announced at the Winter CES, rumors indicated it would simply be an expanded Commodore 64. Even after it was unveiled, some people described it as a CP/M computer with a Commodore 64 emulation mode.

But the Commodore 128 truly is the near-equivalent of three computers in a single box. Outside, all three computers share the same sleek plastic case and 92-key keyboard. Inside, they share the same RAM chips and power supply, but that's about all. The 128 actually contains three separate central processing units (CPUs), two separate BASICs, two independent video display chips, separate banks of Read Only Memory (ROM), and even different memory maps, depending on which mode is selected. What's more, the machine can be operated in five distinct modes: Commodore 64 mode, 128 mode with 40-column video and graphics, 128 mode with text-only 80-column video, CP/M mode with 80-column video, and CP/M mode with 40-column video.

CPU chips include the 6510 for 64 mode and the 8502 for 128 mode—both 6502-compatible—and the Z80A for CP/M mode. Video chips include the VIC-II for 64 mode and 40-column 128 mode, plus an 80-column chip for 80-column 128 mode and CP/M mode. A synthesizer chip—the Sound Interface Device (SID)—is shared by all three microprocessors. Programming languages include BASIC 2.0 in 64 mode and BASIC 7.0 in 128 mode, and a machine language monitor is available in 128 mode and CP/M mode.

Does all this sound confusing? Don't feel bad—it is. Even when using the Commodore 128, you can sometimes forget which mode the computer is in. This is especially true of the 64 mode and 40-column 128 mode, which appear virtually identical on screen.

Furthermore, some modes let you switch to other modes, but not back again without restarting the machine. And speaking of cold-starts, the Commodore 128 can be switched on in *any* of its five modes, depending on its state at power-up. When you hit the power switch, the computer first checks to see if a CP/M system disk is inserted in the drive. If so, it defaults to 40- or 80-column CP/M mode (usually 80 columns). Otherwise, it checks the cartridge slot for a Commodore 64 cartridge program. If it finds one, it automatically switches to 64 mode and runs the cartridge. If there's no 64 cartridge, the computer checks for a 128 cartridge. If it finds one, it comes up in 128 mode (either 40 or 80 columns) and runs the 128 cartridge. If no 128 cartridge is plugged in, the computer checks to see if its 40/80 DISPLAY key is pressed. If so, it starts up in 80-column 128 mode. Otherwise, it switches to 40-column 128 mode.

With so many options, operating the Commodore 128 will take some getting used to.

One of the biggest questions about the Commodore 128 is its degree of Commodore 64 compatibility. Stung by criticism and slow sales because of the Plus/4's lack of 64 compatibility, Commodore went to great lengths to make sure the 128 would run all existing 64 software. In fact, Commodore claims the 128 is 100 percent 64 compatible.

Our tests showed the 128 had no trouble with a wide range of Commodore 64 programs written in BASIC and machine language. We ran a number of programs published in recent issues of COMPUTE! and COMPUTE!'s GAZETTE, including *SpeedScript 3.0*. Only one program failed: "TurboDisk" (COMPUTE!, April 1985). TurboDisk, a machine language utility which speeds up disk loading by as much as 300 percent, ran fine on a 128 hooked up to a 1541 disk drive, but would not work on a 128 connected to the new 1571 disk drive. We weren't particularly surprised, because TurboDisk works by reprogramming the load routines both inside the computer and in the drive. Any drive that isn't completely 1541-compatible cannot handle TurboDisk.

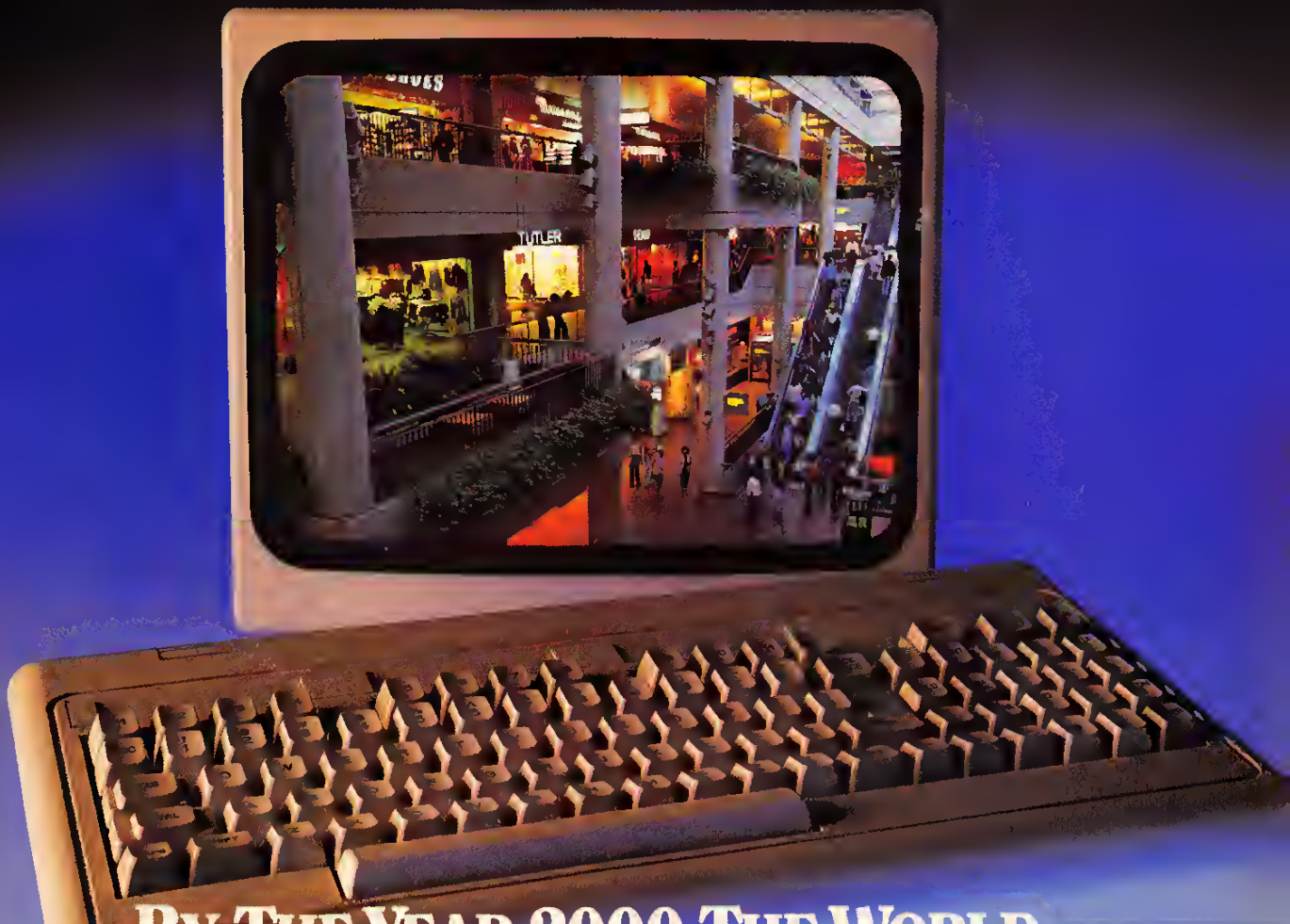
Since the 1571 is designed to be much faster than the 1541, it may seem that utilities such as TurboDisk are superfluous anyway. However, keep in mind that the 1571, like the Commodore 128, is a multimode device. When the 128 is operating in 64 mode, the 1571 drive behaves just like a 1541—it stores 170K of data per disk and runs fairly slow. When the computer is switched to 128 mode, the 1571 speeds up about 500 percent and becomes a double-sided drive, storing about 360K per disk. And when the computer is in CP/M mode, the 1571 runs about 12 times faster than a 1541 and stores 410K on a floppy.

In CP/M mode, the 1571 also is supposed to read disks in IBM System 34 format, such as those made for Osborne and Kaypro CP/M computers. However, the 1571 drive we tested would not load our Osborne disk. A Commodore representative told us the 1571 we were using was still a prototype, and that final production models definitely would read CP/M disks. He also said that production 1571s would be fully 1541 compatible, so that programs like TurboDisk should work too.

If the final 1571s are not completely 1541 compatible, Commodore will run into trouble on another front—commercial copy protection. Some copy-protection schemes depend on precise timing and certain routines within the 1541 disk drive ROMs. If much is changed, the disks won't load. Before acquiring a Commodore 128 and 1571 drive to run 64 software, it would be a good idea to try loading some commercial disks first to make sure they work.

The best news about the Commodore 128 is BASIC 7.0, the powerful BASIC interpreter available in 128K mode. It is, perhaps, the most powerful BASIC ever offered in a personal computer—more complete than even IBM BASIC and MSX BASIC. It contains all the commands in Commodore 64 BASIC 2.0, all the disk and file commands of BASIC 4.0 (as found on the Commodore 8032 and SuperPET), and nearly all the graphics and sound commands of the *Super Expander 64* cartridge, *Simon's BASIC*, and the Plus/4's BASIC 3.5.

BASIC 7.0 makes it possible to draw graphics, define and move sprites, create sound effects, and play music without PEEKs, POKEs, or machine language. Sprite movement is implemented during a machine-level interrupt, so a few BASIC statements can keep up to eight sprites moving simultaneously



BY THE YEAR 2000, THE WORLD MAY CATCH UP WITH THE WAY COMPUSERVE'S ELECTRONIC MALL™ LETS YOU SHOP TODAY.

Presenting the computer shopping service that delivers discount prices, name-brand merchandise, and in-depth product information.

To make your computer even more useful, join CompuServe and shop in our Electronic Mall. Easy enough for beginners, it's open 24 hours a day, 7 days a week. And it offers a wide range of goods and services from nationally known stores and businesses including Bloomingdale's, Waldenbooks, American Express and Commodore.

CompuServe's Electronic Mall™ lets you shop at your convenience in all these departments:

The Auto Shop, Book Bazaar, Financial Mart, Leisure Center, Merchandise Mart, Newsstand, On-line Connection, Personal Computer Store, Record Emporium, Specialty Boutique and Travel Agency.

Take the CompuServe Electronic Mall 15-Minute Comparison Test.

What you can do in 15 minutes shopping the Electronic Mall way.

- Access descriptions of the latest in computer printers, for instance.
- Pick one and enter the order command.
- Check complete descriptions of places to stay on your next vacation.
- Pick several and request travel brochures.
- Access a department store catalog and pick out a wine rack, tools, toys...anything!
- Place your order.

What you can do in 15 minutes shopping the old way.

- Round up the family and get in the car.

The Electronic Mall—A Valuable Addition to the Vast World of CompuServe.

CompuServe Information Services bring you information, entertainment, personal communications and more.

You can access CompuServe with almost any computer and modem, terminal or communicating word processor.

To buy a CompuServe Subscription Kit, see your nearest computer dealer. To receive our informative brochure, or to order direct, call or write:

CompuServe

Information Services, P O Box 20212,
5000 Arlington Centre Blvd , Columbus, OH 43220

800-848-8199

In Ohio call 614-457-0802

The Electronic Mall™ is a service of CompuServe Inc. and L. M. Berry & Company

An H & R Block Company

while the program performs other tasks—or even stops. To make it easier to define sprites, you can grab any predrawn shape off the screen and store it as sprite data, or design the sprite bit by bit with a built-in sprite editor. Playing music with the SID chip has always been tedious because of the large number of POKEs required, so BASIC 7.0 has ten predefined musical instrument sounds available with a single command.

Because BASIC 7.0 is so extensive, we can't explain every command in detail, but we can cover some highlights. Remember that some specifications or syntax rules may have changed by the time the Commodore 128 entered final production.

The disk and file commands include DLOAD and DSAVE (for loading and saving to disk without adding ,8 to the filename); DVERIFY (compare a disk file with a file in memory); CATALOG and DIRECTORY (for displaying disk directories without erasing a BASIC program in memory); COPY (duplicate a file using dual drives); BACK-UP (copy an entire disk with dual drives); APPEND (open a sequential file for updating); COLLECT (reorganize the Block Allocation Map); CONCAT (combine two disk files); HEADER (format a disk); RENAME (assign a new filename to an existing file); SCRATCH (delete a file); DOPEN and DCLOSE (open or close a disk file); DCLEAR (close all disk channels); RECORD (for positioning the relative file pointer); DS and DS$ (read the error channel); BLOAD (load a binary machine language file); BSAVE (save a block of memory as a binary file); and BOOT (load and run a machine language file).

Note that none of these commands adds new capabilities not available with a Commodore 64 and 1541 drive; they merely simplify the syntax. For example, COLLECT is equivalent to OPEN 15,8,15, "V0":CLOSE 15. The commands can also be abbreviated, as in D-SHIFT-L for DLOAD or even SHIFT-RUN/STOP to automatically load and run. In addition, the special function keys are preprogrammed to execute certain frequently used commands, such as DIRECTORY.

Sprite commands not only replace the old-fashioned POKEs, but also offer more options. And the demo programs we saw proved that BASIC 7.0 can move sprites fast enough for good-quality games without machine language.

SPRDEF
Entered in direct mode, this command activates the built-in sprite editor. An editing window appears on screen, and you're prompted to select sprite 1 through 8. A number of subcommands let you clear all the sprite data, move a crosshair, turn pixels on and off, and change colors for multicolor sprites. When you're done designing the sprite, you reenter BASIC by pressing SHIFT-RETURN, then RETURN again. BASIC 7.0 does not require you to

set aside memory for sprite data; instead, it reserves a 512-byte block (for eight 64-byte sprites) beginning at location 3584 ($E00 hex).

SPRITE *#,on/off,foreground, priority,X,Y,mode*
Sets up various sprite attributes, including sprite color, foreground/background priorities, initial X and Y position, and single color/multicolor.

SPRCOLOR
Defines the multicolor registers shared by all sprites.

SPRSAV *sprite#,string*
SPRSAV *string,sprite#*
Moves sprite definition data into a string or vice versa. For example, you can define a sprite by first drawing a shape on the screen with various graphics commands, then copy the shape into a string with the SSHAPE command, and finally move the string into the sprite data block with SPRSAV. You could also copy the sprite pattern directly to the screen with SPRSAV and GSHAPE.

MOVSPR *sprite#,X,Y*
Moves a sprite to the horizontal and vertical screen coordinates specified by *X,Y*. This is called *absolute movement* and is like the POKEs used to move sprites to screen positions on the Commodore 64.

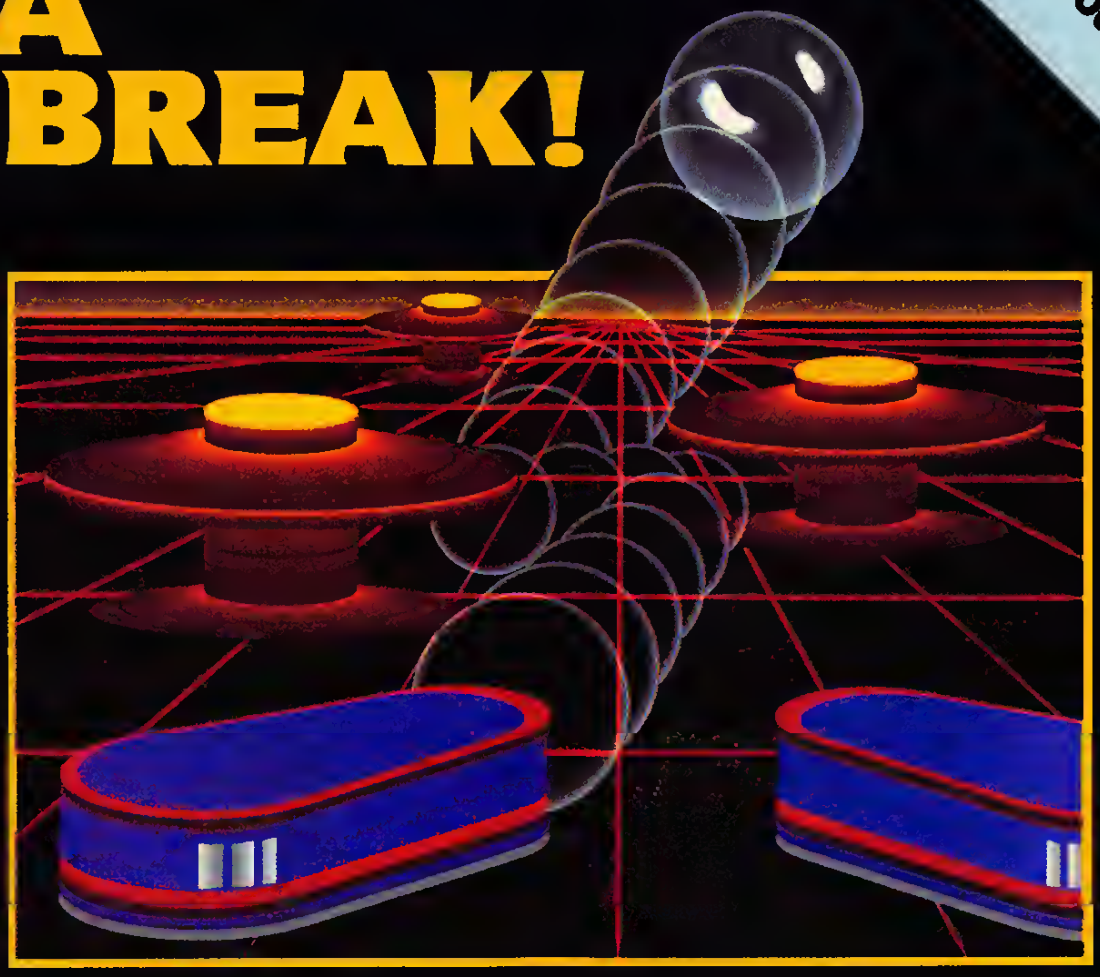
MOVSPR *sprite#,+/−X,+/−Y*
Moves a sprite plus or minus the number of screen coordinates specified by *X,Y*. This is called *relative movement* and is useful when you don't know the sprite's current position. For example, you could move sprite 5 seven positions to the left and ten positions down with MOVSPR 5,−7,+10.

MOVSPR *sprite#,angle #speed*
Moves a sprite continuously in a certain direction at a specified speed via a machine level interrupt, even when the BASIC program is executing other lines or is stopped. When the sprite disappears off the screen, it wraps around and reappears at the opposite end of the screen. This command is useful when you want to send a sprite flying on a predetermined course and speed while your program does other things. For instance, with this command you could quite easily animate the rocks in an *Asteroids*-type game while your program takes care of checking the joystick and moving the player's ship. The *angle* parameter specifies the direction in degrees (0 = up), and *#speed* the velocity. To move sprite 1 continuously along a horizontal path from left to right, you would type MOVSPR 1,90 #5. To move it vertically from top to bottom a little faster, you would type MOVSPR 1,180 #5. Of course, the sprite must be positioned somewhere on the visible screen to begin with.

COLLISION *type,line#*
Detects sprite collisions and diverts the program to a subroutine starting at the line specified by *line#*. The *type* parameter lets you detect different kinds of collisions. Examples: COLLISION 0,1000 passes control to a subroutine at line 1000 when two sprites collide. (The subroutine must end with a RETURN.)

TAKE A BREAK!

For Apple, Atari®, & Commodore 64™



WITH NIGHT MISSION PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

See your dealer . . .

or write or call for more information

Order Line: 800 / 637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

COLLISION 1,2000 passes control to a subroutine at line 2000 when a sprite collides with a screen object. COLLISION 2,10000 diverts the program to a subroutine at line 10000 when a sprite is touched with a light pen.

BUMP(*type*)
Detects sprite collisions and returns a value corresponding to the sprites involved. This makes it possible to determine which sprites collided or if a collision happened off the visible screen (invisible to COLLISION). BUMP(0) records sprite-to-sprite collisions, and BUMP(1) records sprite-to-screen collisions.

Graphics commands make up for the deficiencies in BASIC 2.0 and complement the sprite commands. They're also fairly fast for a BASIC interpreter. Here's just a sampling:

GRAPHIC *mode#,clear,window*
Flips the screen to the graphics mode specified by *mode#*. Mode 0 is 40-column text (default); 1 is hi-res graphics; 2 is hi-res graphics with a text window; 3 is multicolor graphics; 4 is multicolor graphics with a text window; and 5 is 80-column text (RGB only). The text windows are similar to those on Atari and Apple computers—they allow a few lines of text beneath the graphics window on the upper part of the screen. The text windows start at line 19, but that can be changed with the *window* parameter in the GRAPHIC statement. The *clear* parameter lets you specify whether the screen will be cleared upon entering the new mode (0 = no clear, 1 = clear).

COLOR *source#,color#*
Sets up the color registers. The *color#* parameter defines the color from 1 to 16. The *source#* parameter specifies the color register affected—0 for the 40-column background, 1 for the graphics mode foreground, 2 for the multicolor graphics mode primary color, 3 for the multicolor graphics mode secondary color, 4 for the 40-column border, 5 for the character color, and 6 for the 80-column background color.

BOX *source#,X1,Y1,X2,Y2,angle,paint*
Draws a box on the hi-res screen. *Source#* defines the color register (0 to 3), *X1*, *Y1*, *X2*, *Y2* are the X and Y coordinates of the opposite corners, *angle* is the rotation in degrees (default = 0), and *paint* specifies whether the box will be filled (0 = hollow, 1 = solid). Example: BOX 1,10,10,60,60,0,1 draws a filled green box in the upper-left corner of the screen.

CIRCLE *source#,X center,Ycenter, X radius,Y radius,arc angle1,arc angle2,angle,increment*
Draws circles, ellipses, arcs, triangles, octagons, and other polygons on the hi-res screens. *Source#* is the color register (0 to 3), *X center* and *Y center* are the X and Y coordinates of the centerpoint, *X radius* and *Y radius* are the X and Y coordinates of the radius, *arc angle1* is the starting arc angle in degrees (default = 0), *arc angle2* is the ending arc angle in degrees (default = 360), *angle* is the rotation in degrees (default = 0), and *increment* specifies the number of degrees between segments (default = 2). Examples: CIRCLE 1,160,100,65,50 draws a green circle; CIRCLE 1,160,100,65,10 draws a green ellipse; CIRCLE ,60,40, 20,18,,,,45 draws an octagon, and CIRCLE ,260,40,20,,,,,90 draws a diamond.

DRAW *source#,X1,Y1* TO *X2,Y2 ...etc.*
Draws a dot, line, or figure on the hi-res screens. *Source#* is the color register (0 to 3), *X1* and *Y1* are the starting screen coordinates, *X2* and *Y2* are the following screen coordinates, and more coordinates can follow, up to BASIC's line length limit. Examples: DRAW 1,100,50 plots a dot at coordinates 100,50 because no endpoint is specified; DRAW 1,100,50 TO 100,75 draws a vertical line from 100,50 to 100,75; and DRAW 1,10,10 TO 10,60 TO 100,60 TO 10,10 makes a triangle.

LOCATE *X,Y*
Positions the invisible graphics cursor at hi-res screen coordinates *X,Y*. This defines the default starting point for all the drawing commands.

PAINT *source#,Xstart,Ystart, mode*
Fills an area on the hi-res screen. *Source#* is the color register, *Xstart* and *Ystart* define the starting coordinates, and *mode* specifies which area to fill (0 = fill the area defined by *source#*, 1 = fill the area defined by any nonbackground color). Example: CIRCLE ,160,100,65,50: PAINT ,160,100 draws and fills a circle using the default foreground color.

SSHAPE *string,corner1,corner2*
Stores shapes drawn on the hi-res screens into string variables. The rectangular area of screen data between coordinates *corner1* and *corner2* is saved in the string variable *string*. The area which can be stored is limited by the 255-character capacity of a BASIC string. SSHAPE is very similar to GET in IBM BASIC.

GSHAPE *string,corner1,corner2, mode*
Plots the data stored in a string variable as a shape on the hi-res screens. It is the opposite of SSHAPE. *Corner1* and *corner2* define the rectangular screen coordinates, and *mode* specifies how the shape will be plotted. If *mode* = 0, the shape is placed as it exists; 1 inverts the shape; 2 performs a bitwise OR when the shape is overlapped onto the screen; 3 ANDs the shape with the screen; 4 XORs (exclusive-ORs) the shape with the screen. GSHAPE corresponds to PUT in IBM BASIC.

Sound commands in BASIC 7.0 take much of the tedium out of creating sound effects and music with the versatile but complex SID chip. Together, BASIC 7.0 and the SID chip give the Commodore 128 the best sound capability of any personal computer now on the market. Here are a few examples:

SOUND *voice,freq,dur,sweep,min, step,wave,width*
Plays a sound with the selected voice, frequency, and duration. *Voice* can be 1 to 3, *freq* 0 to 65535, and *dur* 0 to 32767 jiffies (a jiffy equals 1/60 second). The following parameters are optional. *Sweep* defines the direction for a sweep (shifting frequencies), with 0 = up, 2 = down, and 3 = oscillation. *Min* sets the minimum frequency for a sweep; *step* picks the step value for a sweep; *wave* chooses a SID waveform (0 = triangle, 1 = saw, 2 = square, 3 = noise); and *width* specifies the width for a pulse waveform.

PLAY *"O oct,T tune,U vol,V voice,X filter,notes"*
Plays one or more notes using a selected octave, envelope, volume,

SUMMER GAMES II.™
EIGHT NEW WAYS TO GO FOR THE GOLD.

Sure Summer Games was great, but why stop there? Let Summer Games II take you even farther with eight new events including cycling, fencing, kayaking, triple jump, rowing, high jump, javelin and even equestrian. They can all be played by up to eight players and some, like cycling, rowing and fencing challenge you with realistic head-to-head competition.

First, you decide which of the 18 different countries you're going to represent. Then, in true Olympic fashion, you will need the proper strategy and mental toughness, not just speed and agility to excel in each event. It's so realistic, there's even an opening and closing ceremony along with medal presentations after each event.

It's not too early to get ready for 1988. With the right diet, proper training and hours of practice you just might make it. In the meantime, put on your sweatsuit, grab that joystick and let Summer Games II give you eight new ways to **Go For The Gold!**

EPYX™
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player

voice, and filter. *Oct* specifies the octave 1 to 6; *tune* an envelope 0 to 9 (see below); *vol* the volume 0 to 9; *voice* 0 to 2; *filter* (0 = off, 1 = on); and *note* can be A, B, C, D, E, F, or G with sharps, flats, dots, and standard durations (quarter notes, half notes, etc.). An M in this parameter tells the computer to wait for all voices currently playing to end. Any number of notes can be strung together in this field, up to BASIC's line length limit. The predefined envelopes selectable with the *tune* parameter are 0 = piano, 1 = accordian, 2 = calliope, 3 = drum, 4 = flute, 5 = guitar, 6 = harpsicord, 7 = organ, 8 = trumpet, and 9 = xylophone.

ENVELOPE #,*attack,decay,sus,rel,wave,width*

Commodore 128 Memory Management And Machine Language

Charles Brannon Program Editor

Using an external memory cartridge, the Commodore 128 can be expanded up to 512K RAM. This memory is not directly available for programs, though, but is used as a *RAM disk*—the functions of a disk drive are simulated with the memory chips. This provides faster throughput than a hard disk, but all information is lost when the power is turned off. You need to dump the contents of a RAM disk to a regular disk at the end of each session.

A special memory management unit (MMU), located at $FF00, controls the 128's complicated memory map. The MMU interprets memory addresses even before the microprocessor sees them. It permits you to swap between banks of 64K, but can leave a small portion of memory as common memory. You don't always want zero page and the stack to disappear when you change banks. The MMU lets you bank between four 64K banks, and allows multiple banks of 256K, up to one megabyte of memory.

The MMU controls whether the VIC chip or 80-column chip controls the screen display, and even senses the position of the 40/80 DISPLAY switch (though the software must interpret this switch). The MMU controls access to RAM or ROM, allowing either to be visible in the memory map. A programmer can set up a series of preset memory configurations and quickly select them by writing to the MMU. The address of the VIC chip can be relocated anywhere within the virtual 256K memory space.

The MMU also controls the fast serial port used with the 1571 disk drive (and conceivably with other fast peripherals). It determines the clock speed of the 8502, and controls which of the three microprocessors (6510, 8502, Z80A) is in control.

Although not supported in ROM, it's possible to have all three microprocessors running by quickly switching between them. Maybe someone will find a way to take advantage of this potential multiprocessing capability.

Machine language programmers will appreciate the Commodore 128's machine language monitor, entered from BASIC with the MONITOR command. It pretends that the 128K of memory is contiguous and permits five-digit hexadecimal addresses. It makes full use of 80 columns if selected. The monitor works much like *64 Supermon*, with commands to assemble, disassemble, fill, go to address, hunt through memory for a hexadecimal string, load, display memory with ASCII equivalents, display registers, save, transfer a block of memory, verify a saved program, exit to BASIC, modify memory, modify registers, and display disk error status.

BASIC commands for machine language include BLOAD and BSAVE to load and save machine language programs or other binary files, and BOOT to load and run a machine language program. The familiar USR, WAIT, POKE, PEEK, and SYS commands can now reference the second 64K of memory with the BANK command. SYS can be followed by four parameters that are transferred into the accumulator, X register, Y register, and status flag register. After a SYS, the RREG command can transfer the contents of these registers into four variables. This makes it much easier to pass information between BASIC and ML.

The 8502 microprocessor in 128 mode is opcode-compatible with the 6502 and 6510, but can now run at two megahertz, twice the speed of the VIC-20's 6502 and Commodore 64's 6510. All VIC/64 Kernal routines are supported, making program translation much easier. New Kernal routines support special features of the 128, including routines for memory management.

A RESET button near the power switch can coldstart the machine. Holding down RUN/STOP with RESET initiates a "lukewarm" start. It's a more thorough reset than RUN/STOP-RESTORE, but still retains your BASIC program. This reset puts you into the machine language monitor, where you can exit back to BASIC with no harm done.

YOUR COMMODORE 64™ CAN NOW USE STANDARD APPLE™ II+ HARDWARE AND SOFTWARE

WITH THIS

At Mimic we believe that you and your computer should dictate the choices of hardware and software you can use.

The Spartan™ was developed to allow you to choose the hardware and software that best suits your needs.

Our goal in designing the Spartan™ was simple. To take what you already have and give you more.

Mimic Systems is proud to give you the Spartan™ The Apple™ II+ emulator for the Commodore 64™

Spartan™ Suggested Retail Prices:

The Spartan™ (includes BUSS, CPU, and DOS cards) $599.00
BUSS card $299.00
CPU card (requires BUSS card) $199.00
DOS card (requires BUSS and CPU card) $199.00

(All prices in U.S. Funds Freight not included.)
American Express, Visa and MasterCard accepted

MIMIC

FOR INFORMATION WRITE:

MIMIC SYSTEMS INC.
1112 FORT ST., FL. 6N
VICTORIA, B.C.
CANADA V8V 4V2

To Order Call:
1-800-MODULAR
(663-8527)

Redefines any of the ten predefined music envelopes for the *tune* parameter of the PLAY command. The # specifies the envelope (0 to 9), followed by the values for attack, decay, sustain, and release. *Wave* sets the SID waveform and *width* selects the width of a pulse waveform.

FILTER *freq,lopass,bandpass, hipass,res*
Switches the SID filters for use with the *filter* parameter of the PLAY command. *Freq* selects the frequency; *lopass* the low-pass filter (0 = off, 1 = on); *bandpass* the notch-reject filter (0 = off, 1 = on); *hipass* the high-pass filter (0 = off, 1 = on); and *res* the resonance (0 to 15).

Despite the almost bewildering array of commands listed above, we've barely scratched the surface of BASIC 7.0. Indeed, a preliminary manuscript for the *Commodore 128 System Guide* is a stack of single-spaced, typewritten pages two and a half inches thick.

There are commands for windowing, switching 64K memory banks, renumbering BASIC programs, deleting ranges of BASIC lines, assigning new definitions to the predefined special function keys, entering the machine language monitor, trapping runtime errors and diverting execution to an error-handling routine at a certain line number, resuming execution after a runtime error, highlighting errors in BASIC lines, constructing loops without FOR-NEXT, and inserting delay loops. Plus additional commands for sprites, sound, music, and graphics that we didn't have room to mention.

Commodore BASIC 7.0 is a predictable step in the evolution of high-level programming languages for personal computers. It continues the trend away from low-level instructions such as PEEK and POKE—vestiges of machine language—and further shields users from intimate contact with the bits and bytes of computer circuitry. Yet, unlike some other personal computers introduced in recent years, the Commodore 128 retains its BASIC as a built-in feature and also provides a machine language monitor for those who want to explore the computer at every level. It's a welcome combination. ©

NEW LOW PRICES FROM THE WORLD'S LARGEST SPORTS SOFTWARE COMPANY!

Whether you're a coach, player, computer bug, statistician, dabbler, doer or just a real sports fan, PDS has a package for you...at our lowest prices ever! Check our line-up below, and then order your first PDS Sports Package.

FOOTBALL STATISTICS. A comprehensive computer software system. You compare teams in over 60 offensive and defensive situations and watch the matchups on the screen. Every team is given a power rating. PDS also provides "year-to-date" data diskettes for a nominal charge. 39.95

FOOTBALL HANDICAPPING. (Includes FOOTBALL STATISTICS Package). Forecasts the outcome of football games based on massive amounts of statistical data, morning lines and point spreads. 49.95

BASKETBALL STATISTICS. The most comprehensive basketball program ever written. "What-if" your way into every imaginable comparison and formulation. 39.95

BASKETBALL HANDICAPPING. (Includes BASKETBALL STATISTICS). Gives you an inside look into the outcome of games by mathematically equating statistics, lines and point spreads. 49.95

BASEBALL STATISTICS. See how opposing baseball teams stack up against each other. See summarizations. Performance stats on every team in the major leagues. 39.95

BASEBALL HANDICAPPING. (Includes BASEBALL STATISTICS) Makes you better-prepared to predict the winner of any game. At least, mathematically. 49.95

THOROUGHBRED HANDICAPPING. This "world class" program gives you an edge in predicting winners. A proven system...by PDS and many horseracing enthusiasts. 129.00

HARNESS RACE HANDICAPPING. The fast, easy way to find out what races and what horses should be looked at...all summarized in a "power rating" format. 129.00

QUARTER HORSE HANDICAPPING. the computer system actually "rates" the horses in each race of what is the fastest growing segment in the sport of kings. 129.00

TRAINER STATISTICS. Analyzes all local racehorse trainers and gives you a rating for each one. 39.95

JOCKEY STATISTICS. A short time with this software package will show you why the top 15 jockeys at local tracks win over 90% of all races. 39.95

HARNESS DRIVER STATISTICS. A complement to the HARNESS HORSE HANDICAPPING System, this package tells you all you need to know about the man (or woman) *behind* the horse. 39.95

PDS SPORTS™

P.O. BOX E / TORRANCE, CA 90507 / (213) 516-6688
Please send me the following PDS SOFTWARE PACKAGE:

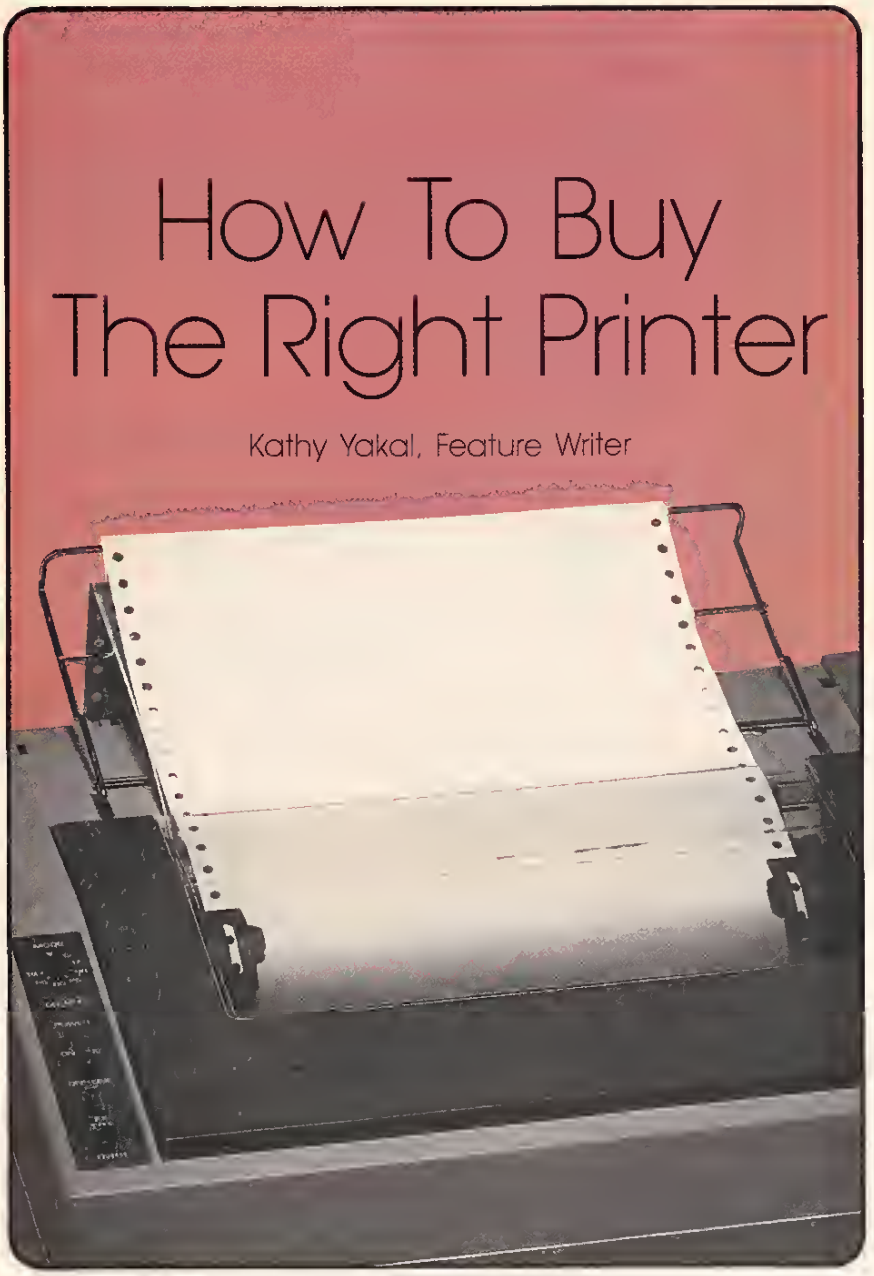
_____...$_____
_____...$_____
_____...$_____
Total

Add $6.00 for postage and handling.
Calif. residents add 6½% sales tax
I have an APPLE ( ) IBM ( ) TRS-80 ( ) COMMODORE-64 ( )
DISKETTE ( ) CASSETTE ( ); MODEL # _____
( ) Check enclosed
( ) Charge my credit card: A E. ( ), VISA ( ), M.C. ( )
Card No._____ Expires_____
Signature_____
(As it *appears on credit card*)
OR CHARGE BY PHONE...CALL (800) 222-2601
(In Calif: (213) 516-6688)
NAME_____
ADDRESS_____ PHONE_____
CITY_____ STATE_____ ZIP_____

Dealer Inquiries Invited

How To Buy The Right Printer

Kathy Yakal, Feature Writer



Choosing a printer may be the most difficult decision you'll make when assembling your computer system. Even if you're planning to use it only for personal letters and program listings, there are more alternatives to consider than with any other peripheral.

Once you buy a computer, selecting a tape or disk drive and a monitor isn't too tough. There aren't that many choices. Modems are a bit more difficult because of the number available, but their features and performance don't vary all that much.

Printers, however, are another story. There are dozens of printers which are inexpensive and produce good quality print. And they're complicated pieces of machinery. "As opposed to a computer, which has relatively few mechanical or moving parts, printers are the biggest mechanical piece of your computer system," points out Dave Crowley, corporate communications specialist for Centronics.

"Consumers have to be aware that if there is going to be a problem, it probably will happen in the printer rather than anywhere else in the computer system."

Further, everyone's needs are different. Some people just want to print out program listings. Others want to print school papers or personal letters or business correspondence, or pictures created with light pens or graphics tablets. The goal is to find a printer that has everything you need without spending lots of extra money for features you'll never use.

The first step in buying a printer, then, is to determine exactly what you need. "We always recommend that someone sit down and draw up a list of items that the printer will be used for, and take that with them to the dealer," says Crowley.

Here's a list of questions, compiled with the help of Crowley and Star Micronics Marketing Director Tom Bongiorno, that you might want to ask yourself before visiting computer dealers:

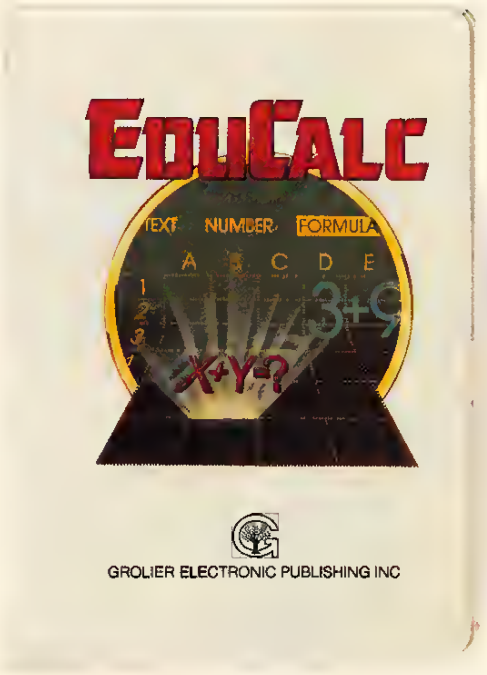
• *Will this printer work with my computer?* And if it doesn't, is the dealer knowledgeable enough to help you figure out how to make it work? Compatibility should be assured.

Printers for personal computers are designed to work with two general types of interfaces: *parallel* and *serial*. For the purposes of hooking up a printer, either works just as well. The most common parallel interface is often referred to as a Centronics-standard interface, and the most common serial interface is called the RS-232. If your computer doesn't have one of these interfaces built in, you may need to buy an interface adapter in addition to a printer. In any case, make sure the printer you buy is compatible with the interface available on your computer.

Many printer manufacturers offer cables that connect directly to, say, a Commodore 64 or an Atari. Buying such a printer frees you from compatibility worries for now, but may also restrict its future use if you someday buy a different system. Some printers allow interfacing through either a standard or a machine-specific port; these provide greater flexibility for use with other computers.

Consider software compatibility, too. Even if you're going to use

Modern Tools for Modern Minds...



Grolier presents the first and only series of productivity tools with a built-in tutorial and practice.

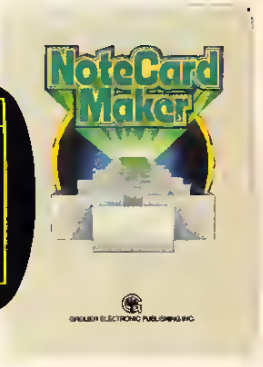
NEW! EduCalc™

...the first electronic spreadsheet program that also teaches beginning users to understand and use a spreadsheet!

- Self-paced, menu-driven tutorial lets you determine the amount of help you need at any one time, and then return later for additional instruction.
- Practice template, for creating a personal budget, enables you to become familiar with a standard spreadsheet before developing your own.
- No need to memorize commands; directions appear on each screen.
- Makes it easy to create, edit, save and print a spreadsheet up to 26 columns wide by 99 rows deep.
- Enter text, numbers or formulas, define constants, and sort information alphanumerically.

...there's never been a more simple solution for creating budgets, charts, schedules and tables for school and home use!

ONLY $49.95 Suggest Retail Price



NEW!

Friendly Filer™

...a great database management learning experience that's also great fun!

- Teaches database management through the use of an entertaining tutorial and stimulating questions.
- Researches the answers by searching out and sorting a built-in "animals" database.
- Uses simple, self-explanatory menus, to lead the user step-by-step into creating files by entering, selecting, sorting and printing data.

...a marvelous learning experience that stresses imagination, creativity and fun. And it's productive, too!

ONLY $39.95 Suggest Retail Price

Note Card Maker™

This remarkably powerful program is a clever tutorial... a helpful tool...and a terrific way to take a lot of the time and effort out of preparing bibliographies and note cards.

- Interactive, self-paced tutorial demonstrates exactly how to prepare note cards.
- Provides user with a standard format for gathering research information, including key word, source code and notes.
- Uses the database management facility to sort and search for data...then incorporates the word processing capabilities to reorganize, select, edit and print-out note cards in any desired sequence.

ONLY $39.95 Suggest Retail Price

Available for the Apple® II family of computers. Also available for the Commodore 64™ and IBM® PC and PCjr. in Fall 1985.

See your computer software dealer today. Or call Grolier Electronic Publishing Toll-Free. 800-858-8858.

Grolier Electronic Publishing, Inc.

SHERMAN TURNPIKE, DANBURY, CONNECTICUT, 06816 • (800) 858-8858

Legend, The Clear Choice.

How Do You Know You Chose The Right Printer?

You can choose from several types of printers. They're available from more than 50 manufacturers. With and without graphics. In high speed and low speed models at prices ranging from less than $200 to well over $2,000.

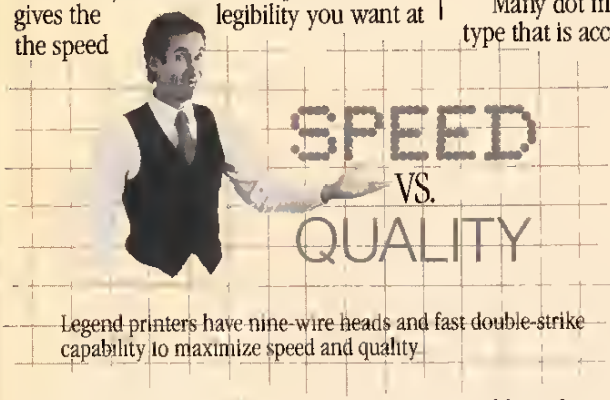
How do you know you made the right choice?

Here's some easy-to-understand facts from Legend to help you make a "clear choice."

Legendary Legibility.

There are trade offs in buying printers. Simply stated, within a given price range, quality, or legibility, decreases as speed increases.

The object is to find the printer that gives the legibility you want at the speed you need and at the price you can afford, like a Legend.



Legend printers have nine-wire heads and fast double-strike capability to maximize speed and quality.

Dot matrix type printers are the most popular and lowest cost printers. Most combine high speed with acceptable quality and legibility. They're extremely versatile and very dependable.

Legendary Head.

Dot matrix printers have print heads containing tiny pins that "fire" against a ribbon to make a series of dots that combine to form letters, numbers and graphics. Generally, the more pins or "wires," the closer together the dots, and the better the legibility.

Legend printers have "full nine wire" heads for better legibility.

Many dot matrix printers produce type that is acceptable for about 95% of all correspondence—invoices, letters, and the like.

Daisy Wheel or "letter quality" printers run one fifth as fast and cost twice as much as a Legend. So a Legend dot matrix printer makes better sense. Why? Read on.

Easy to Switch.

Many dot matrix printers have a "double strike" capability that reduces the speed, but produces better legibility.

Unfortunately, with most dot matrix printers, changing to the double strike mode is difficult. And, unlike Legend, most other printers only run at 25% of their normal speed.

Legend printers have a special, easily accessible switch on the top of the machine so double strike capability (Legend calls it "damn near letter quality") is at the operator's fingertips. And machine speed stays at a very productive 50% of normal speed.

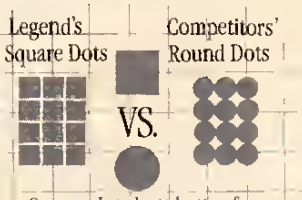

Square dots are 25% bigger than round dots.

Square Vs. Round.

In addition to speed, the shape of the dot affects the legibility of type, too.

Most printers use round dots. Legend printers use square dots because they butt better and fool the eye into thinking that lines are continuous.

Think of it this way. Imagine you stack a series of baseballs next to a series of equal sized blocks. Now move back 20 paces and look at the two stacks. Which one would look most like a straight line?

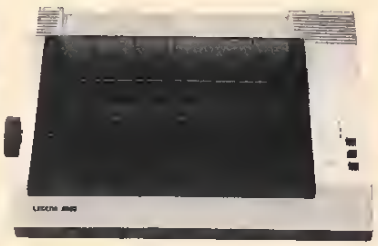


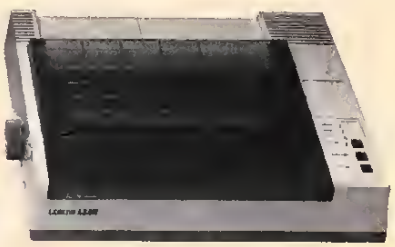
Square dots butt better for higher legibility.

Legendary Graphics.

A picture is indeed worth a thousand words. And today's sophisticated software packages are making it easier to translate data into graphics that communicate quickly and clearly.


Legend 880—100 cps/80 col


Legend 1080—140 cps/80 col


Legend 1380—160 cps/80 col

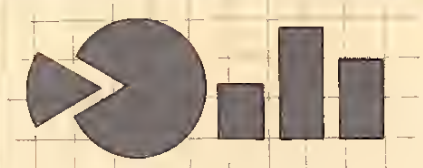
Legend 1385—160 cps/136 col

Unfortunately, not all printers are capable of running graphics software, including some of the more popular models. If your application includes charts, graphs or other kinds of symbols, it's important that you pick a printer that is compatible with the software and capable of printing graphics, like Legend.

Legend printers are compatible with almost all popular graphics software programs. What's more, you'll get more characters built into memory when you choose a Legend.

More Graphics.

Most comparably priced printers feature 96 to 196 characters (symbols) built into memory. Legend printers have 228,



Legend printers have 228 characters in memory to produce more graphics and more languages

so you can produce more graphics and more languages (French, German, Greek, Spanish, Italian) so you can be more productive. And isn't that the bottom line, really?

Legend's advanced square dot technology will make your charts and graphs look sharper, too.

A Head For Life.

No printer at any price is worth its salt if it's not dependable. Legend's square dot heads use a special alloy that maintains a sharper image and a longer life. So all Legend printers come with a lifetime head warranty.

If anything ever goes wrong with the head, simply send it back for an immediate exchange. It's so simple, it's legendary.

Legendary Value.

If there's still any doubt in your mind about which printer is best for you, we'll make your decision even easier.

Legend printers also come with standard friction and tractor feed and are compatible with most computers.* They feature over 40 software selectable type styles and make a crisp original plus three copies.

LEGENDS LIVE FOREVER! UNLIMITED HEAD WARRANTY

Most remarkable of all, prices start at just $279. *And for just $1 you can get a special buffer upgrade.* Compare if you like, but we'll bet you'll find Legend Printers, feature-for-feature, to be clearly the best value for your money today.

Call *1-800-4-LEGEND* today for more information and visit your dealer to see them in action.

Dealer inquiries call 1-800-321-4484. In CA call (818) 704-9100. Or write CALABCO Peripherals Division, 6041 Variel Avenue, Woodland Hills, CA 91367.

Legend and Legend Peripheral Products are trademarks of Cal-Abco

LEGEND

The Clear Choice.

*Legend Printers can be interfaced with most computers, including Apple® II, IIe, IIc, Atari,® Columbia® 1600 series, Compaq,™ Commodore® Compaq Plus,™ DeskPro,™ Eagle® PC™ and Spirit,™ Turbo,™ Hewlett Packard® 110, 150, IBM® PC, XT, AT, Kaypro,® Sanyo,® TI® Professional, TRS®

the printer only to print out correspondence and home records, make sure the printer will accept your word processor's formatting commands.

• *How good does your printed copy need to look?* Different kinds of printers offer varying type qualities.

Impact printers (*daisy wheel* and *dot-matrix*) strike the paper through an inked ribbon to form characters and graphics. Daisy wheel printers are capable of producing *letter-quality* type because the characters are stamped onto the paper by a print wheel that works much like the strikers on a typewriter. Dot-matrix printers, on the other hand, have printheads with dozens of tiny metal pins that reproduce characters and graphics by printing tightly packed patterns of dots. Some dot-matrix printers, depending on the numbers and placement of pins, can produce near letter-quality print.

Thermal printers don't use an ink ribbon—they burn characters and graphics onto special paper coated with a heat-sensitive film. They're relatively inexpensive and quiet compared to other printers, but the special paper costs more. *Thermal transfer* printers, however, can print on any kind of paper.

Ink-jet printers, the third kind found in the under-$1,000 price range, spray ink through tiny holes to reproduce characters.

"You should always look at the output of the printer—get a demo—because draft mode or near-letter quality from three different printers will always differ," says Bongiorno. "If you like the typeface on one printer better than another, then to you the quality will be better, and that's the one you should buy."

• *Are there hidden costs?* Printers that require special paper or additional interface cables can make that under-$300 printer move closer to the $400-plus range. If the printer uses a ribbon, what is its average life? Can it be re-inked?

• *What type of paper handling does the printer use?* Printers can move paper around the platen in two different ways. *Friction-feed* printers work like typewriters; they grip one sheet at a time and roll it through. *Tractor-feed* printers use a pair of cogged wheels to grip holes punched into the edges of special printer paper. The paper comes as a long, continuous form with perforated pages, and the holes along the edges are also perforated for removal. The tractor wheels can be adjusted to accommodate different paper sizes and special forms, such as mailing labels. (A printer with non-adjustable tractors is called a *pin-feed* printer.)

Some printers offer both friction- and tractor-feed by including snap-on tractors. This gives you the option of printing correspondence on letterhead stationery, or printing continuous forms.

• *How easy is it to obtain parts?* Daisy wheels, for example, sometimes break after heavy use. Can you run down to a local office supply store to replace the wheel, or will you have to order it by mail and wait six weeks for delivery? If the printhead on a dot-matrix printer burns out, can it be replaced without exchanging the entire printing mechanism? How economical are replacement ribbons? If you buy a thermal printer, is the special paper readily available?

• *How good is the warranty?* Opinions vary on what constitutes a fair warranty period, but, in all cases, find out what it is. Bongiorno believes that one year is fairest. "If something is going to go wrong, it will go wrong within the first year," he says. "Sometimes the warranty on the printhead is different than the printer, which you have to be careful of."

• *Can the printer produce the kind of graphics you need?* You may not need graphics at all, of course, but if you do, there are basically two types of graphics that printers can produce. With *block graphics*, the printer recognizes a block of dots as a symbol or special character. It constructs the picture by assembling these symbols. With *dot-addressable graphics*, each tiny dot can be individually printed, so the printer isn't limited to a set of predefined block graphics patterns. If you need to reproduce high-resolution drawings, charts, or diagrams, you'll probably need a printer with dot-addressable graphics. It will also have to be a dot-matrix printer, because daisy wheel printers aren't capable of producing detailed graphics.

Although these are the major concerns, there are myriad other questions you'll want answered when shopping for a printer:

How fast is it? Printer speeds are usually measured in characters per second, abbreviated cps. Dot-matrix printers are generally much faster than daisy wheel printers in the same price range, but remember that speeds can vary in different printing modes.

How noisy is it? For home use, this might be critically important if your computer shares a family room with the TV.

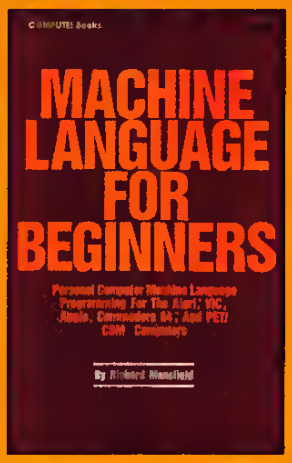
Does it have a buffer? How large? A printer buffer is an area of memory inside the printer that lets it print your material while freeing up the computer for other tasks. To be useful, a printer buffer should be at least as large as the files you need to print. For instance, if you'll frequently be printing documents about 16K long—roughly eight double-spaced pages—the printer buffer should contain at least 16K of memory. To keep initial costs down, you can often buy a printer without a buffer and add the memory later. Also, printer buffers are available as separate devices that connect between a computer and any standard printer.

Can the printer produce special type styles, like italics, boldface, condensed, expanded? Keep in mind that even though a printer offers special fonts, your software may not be capable of sending the proper codes to take advantage of these features. This is a tricky problem that can only be solved by carefully comparing the software manual with the printer manual.

Can I afford it? The bottom-line question.

A good computer dealer, says Crowley, will take the time to answer all of your questions and run the printer through its paces, just as a good car dealer will take you out for a test drive. "I've actually gone in to buy a car and gotten a salesman who couldn't tell me whether it had six or eight cylinders," he says. "If you run up against somebody who's not willing to take the time, then take your business someplace else—because there are plenty of people who are willing." ©

COMPUTE! Books brings you the companion volume to the best seller, *Machine Language for Beginners*, about which the critics have said:



"If you know BASIC and want to learn machine language, thie ie the place to start . . . . Building on your experience as a BASIC programmer, Manefield very gently takes you through the fundamentals of machine language."—Whole Earth Software Catalog

"Underetandable"—The New York Times

"Preeente the machine language novice with a very good tutorial in simple, underetandable terme."—Antic

"I highly recommend Machine Language for Beginnere as your firet introduction to the world of machine language."—Commodore Power/Play

The Second Book of Machine Language for the Commodore 64, VIC-20, Apple, Atari, and PET/CBM

The Second Book of Machine Language picks up where *Machine Language for Beginners* left off. This new book contains one of the most powerful machine language assemblers currently available. The LADS assembler is a full-featured, label-based programming language which can greatly assist you in writing machine language programs quickly and easily.

It's also a clear, detailed tutorial on how large, complex machine language programs can be constructed out of manageable subprograms.

There are powerful computer languages and there is good documentation, but rarely has a sophisticated language been so completely documented as it is in this book. When you finish with this book, you'll not only have a deeper understanding of machine language—you'll also have one of the most powerful machine language assemblers available. And since everything is thoroughly explained, you can even add custom features to the assembler to create a custom language that does just what you want it to (the book shows you precisely how to modify the assembler).

For Commodore 64, Apple (II, II+, IIe, and IIc, DOS 3.3), VIC-20 (8K RAM expansion required), Atari (including XL, 40K minimum), and PET/CBM (Upgrade and 4.0 BASIC). Disk drive recommended.

15% Discount
Buy both books for $25.00 and save $4.90! That's 15% off the retail price!

Offer Expires July 15, 1985.

THE *LADS* Disk

LADS, the assembler used in *The Second Book of Machine Language*, is available on disk for only $12.95. This is a great accompaniment to the book, saving you hours of typing time by providing the complete source and object modules for all versions of the assembler. And LADS disks are specific to your Apple, Atari, or Commodore computers.

To Order: Call Toll Free 800-334-0868 (in NC 919-275-9809) or mail this coupon with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403. Offer Expires July 15, 1985.

_____ *The Second Book of Machine Language*, **$14.95**
_____ *Machine Language for Beginners*, **$14.95**
_____ *LADS Disk* (Apple) **$12.95**
_____ *LADS Disk* (Atari) **$12.95**
_____ *LADS Disk* (Commodore), **$12.95**

☐ Payment Enclosed (check or money order)
☐ Charge ☐ MasterCard ☐ Visa ☐ American Express

Acct. No. _____

Signature _____

Name _____

Address _____

City _____

State _____ Zip _____

_____	1 Book for	$14.95
_____	2 Books for	$25.00
_____	*LADS Disk* for	$12.95
NC residents add 4.5% sales tax		$_____
Shipping and handling ($2.00 per book $1.00 per disk)		$_____
Total Paid		$_____

All orders must be prepaid
Please allow 4–6 weeks for delivery

COMPUTE! Publications, Inc. One of the ABC Publishing Companies

756111C

Solving Common Printer Problems

Selby Bateman, Features Editor

Few things in computing are as frustrating as a recalcitrant printer. Here are some tips on how to find relief.

At one time or another, every computer user looks at the paper rolling out of a printer and sees something that seems to have been sent from an alternate universe. *That's* not what I told my computer to print!

Your neatly formatted double-spaced letter is being printed all on one line. Or your beautiful four-color screen illustration is appearing on paper as a series of capital E's. The italics and underlining you've added for emphasis in a report have changed the rest of the words to an unknown foreign language. Or, perhaps most depressing, the paper is simply rolling out of your printer completely blank.

Nine times out of ten, your printer problems won't be mechanical in nature. More likely, they'll fall into one of two major areas, which we may call *interface/configuration* mistakes and *special effects* errors.

Problems with interfacing and configuring your computer and printer usually happen during your first attempts to connect everything together. But confusion over special effects—such as boldfacing, underlining, super- and subscripting, and graphics—can happen even to the most advanced computer user.

No matter what the cause of a printer problem, it is a frustrating experience. Yet, with some patience and a thorough understanding of how the computer, printer, software, and printer interface work together, you can unleash all of the power and high-quality performance packed into today's printers.

Whether your computer is a Commodore, IBM, Apple, Atari, TI, or other brand, you should become familiar with how it connects to a printer. Often an extra interface is required to allow an otherwise incompatible printer and computer talk to each other.

A thorough discussion of the many printer interfaces for microcomputers could fill a book. But basically, your data will be sent from computer to printer either in a *serial* or *parallel* stream, one bit at a time or eight bits at a time. Most printers use the parallel method. Your computer and printer manuals will tell you which kind of interface to use. But you should also know that some computers require additional accessories to work with certain printers.

For example, the Apple II needs an interface cable and either a parallel or serial interface card. IBM PCs need either the standard printer interface card for parallel connection or an asynchronous serial card. A Commodore 64 can hook directly to Commodore printers to make use of the special graphics symbols and reverse-video characters, but if you want to print special character sets, different type fonts, or foreign language characters, you'll need other printers and appropriate interfaces. Similarly, Atari computers hook up directly to Atari printers, but require the 850 Interface Module or a substitute to work with other printers.

That super-low-priced printer might not look like such a bargain when you arrive home and find that you not only need an additional $35 cable but also a $100 add-on interface. Although most stores selling printers have salespeople to help answer your questions, you should still do your homework with computer manuals, magazines, and books.

To add to the confusion, the application software you want to use—such as a word processor or graphics program—can add its own complications. Unless you configure your system correctly, what you end up with may be quite different from what you want.

For example, let's say your interface automatically sends a *linefeed* instruction which tells the printer to advance the paper. Your word processing program may already contain a similar command. And the printer, unless adjusted, may automatically add a linefeed as well. As a result, when you try to print out a single-spaced letter, the printer may be following instructions to put two or three linefeeds between each line of print. Conversely, you could also end up with no linefeeds at all. The entire letter might be printed on a single line.

The solution, of course, is to enable or disable the linefeeds, depending on the problem. This may involve opening up the printer or interface to flip a switch, or issuing the appropriate command with the word processor program. The answers are buried somewhere in the manuals.

Once you've got the printer and computer connected properly, you'll eventually want to take advantage of the advanced options which printers now offer. The special effects which turn your system into so much more than a typewriter are *theoretically* quite easy to control. The complexity stems, once again, from all the configuration possibilities. Versatility has a price.

Let's consider an example using the *SpeedScript 3.0* word processor recently published in COMPUTE! for Commodore, Atari, and Apple computers. To underline a word with

MORE SSI WARGAMES COMING YOUR WAY.

BREAKTHROUGH IN THE ARDENNES™ is truly one of the finest simulations of the Battle of the Bulge. As battle commander, you can build up or break down regiments and divisions—allocate corps artillery, conduct airstrikes, move and fortify units, and change unit mode. You supply your troops along the extensive road network of the Ardennes.

APPLE[1] ATARI[2] C-64[3] $59.95.

On September 17, 1944, Montgomery launched a massive airborne/ground assault to cross the great rivers of Holland into Germany. Historically, the operation failed. Our OPERATION MARKET GARDEN™ is a gaming success. From the authors of the award-winning "Knights of the Desert™" Contains both Advanced and Intermediate Games.

APPLE[1] ATARI[2] C-64[3] $49.95.

COMPUTER MAIL ORDER

THE CMO ADVANTAGE

- THE BEST PRICES! We will meet or beat any qualified price you find.
- Next day shipping on all in stock items
- Free easy access order inquiry.
- Orders from outside Pennsylvania and Nevada save state sales tax.
- Free technical support with our factory trained technical staff.
- There is no limit and no deposit on C.O.D. orders
- There's no extra charge for using your credit card. Your card is not charged until we ship.
- We accept purchase orders from qualified corporations. Subject to approval
- Educational discounts available to qualified institutions
- FREE CATALOG MEMBERSHIP

ORDER LINE

1-800-233-8950
In PA 1-800-242-4215

CUSTOMER SERVICE AND TECH SUPPORT
1-717-327-1450

MAILING ADDRESS

EAST
Dept. A206, 477 E. Third St.
Williamsport, PA 17701

WEST
Dept. A206, P.O. Box 6689
Stateline, NV 89449

MEMBER DIRECT MARKETING ASSOCIATION

CREDIT CARDS

VISA MasterCard AMERICAN EXPRESS DINERS CLUB

SHIPPING

Add 3%, minimum $5.00 shipping and handling on all orders Larger shipments may require additional charges.

All items subject to availability and price change

Returned shipments may be subject to a restocking fee

CANADIAN ORDERS

1-800-268-3974
Ontario/Quebec

1-800-268-4559
Other Provinces

1-416-828-0866
In Toronto

TELEX: 06-218960

2505 Dunwin Drive,
Mississauga, Ontario
Canada L5L1T1

All prices shown are for U.S.A. orders.
Call The Canadian Office for Canadian prices

HOME COMPUTERS

APPLE

APPLE IIe	CALL
APPLE IIc	CALL
MacINTOSH	CALL
IIc LCD Display	CALL

ATARI

65XE (64K)	NEW CALL FOR PRICES
130XE (126K)	
1308T (128K)	
5206T (512K)	

600XL & 600XL CALL WHILE SUPPLIES LAST

850 Interface	$109.00
1010 Recorder	$49.99
1020 Color Printer	$79.99
1025 Dot Matrix Printer	$199.99
1027 Letter Quality Printer	$289.99
1030 Direct Connect Modem	$89.99
1050 Disk Drive	$179.99
Touch Tablet	$64.99
7097 Atari Logo	$74.99
4018 Pilot (Home)	$57.99
5049 VisiCalc	$59.99
CX30 Paddles	$11.99
CX40 Joystick	$7.99
4011 Star Raiders	$12.99
4022 PacMan	$16.99

BOARDS FOR ATARI

Axlon 32K	$39.99
Axlon 48K (400)	$69.99
Axlon 128K	$269.99
Microbits 84K (800)	$109.00
Bit 3 Full View 80	$229.00

commodore

NEW Commodore 128, LCD..CALL

SX-64 Portable	$499.00
Commodore Plus 4	$199.00
CBM 64	$149.00
C1541 Disk Drive	$199.00
C1530 Datasette	$39.99
M-801 Dot Matrix Printer	$189.00
M-802 Dot Matrix/Serial	$219.00
MCS 803 Dot Matrix	$179.00
C1802 Color Monitor	$199.00
C1660 Auto Modem	$59.99
DPS 1101 Daisy Printer	$339.00

Professional Software

Fleet System II w/Spell	$49.99

pfs:

File (64)	$59.99
Report (64)	$49.99

Precision Software

Superbase 64	$54.99

BATTERIES INCLUDED

PaperClip w/Spell Pack	$79.99
The Consultant DBMS	$69.99
Bus Card II	$139.00
80 Col Display	$139.00

commodore

CBM 8032	$839.00
CBM 4032	$599.00
2031 LP Disk Drive	$299.00
8050 Disk Drive	$949.00
8250 Disk Drive	$1249.00
4023 Printer	$329.00
8023 Printer	$589.00
8400 Printer	CALL
Z-RAM	$299.00
Silicon Office	$299.00

Professional Software

Word Pro 4 Plus/5 Plus each	$239.00
Info Pro	$179.00
Administrator	$399.00
Power	$69.99

PORTABLE COMPUTERS

HEWLETT PACKARD

41CV	$189.99
41CX	$249.99
HP 71B	$419.99
HP 11C	$62.99
HP 12C	$89.99
HP 15C	$89.99
HP 16C	$89.99
HP 75D	$999.99
HPIL Module	$98.99
HPIL Cassette or Printer	$359.99
Card Reader	$143.99
Extended Function Module	$63.99
Time Module	$83.99

We stock the full line of HP calculator products

NEC

PC-6401	$749.00
PC-8201 Portable Computer	$299.00
PC-8231 Disk Drive	$599.00
PC-8221A Thermal Printers	$149.00
PC-8281A Data Recorder	$99.99
PC-8201-06 8K RAM Chips	$105.00

SHARP

PC-1350	$159.99
PC-1261	$159.99
PC-1260	$109.99
PC-1500A	$185.99
PC-1250A	$88.99
CE-125 Printer/Cassette	$128.99
CE-150 Color Printer Cassette	$171.99
CE-161 16K RAM	$134.99

DRIVES

HARD

ALLOY

PC Stor	CALL
PC-Disc	CALL
PC QIC-Stor	CALL
PC Back-Up	CALL

EVEREX

Hard Drives	CALL
Tape Back Up	CALL

IDEAssociates

5 meg Removable/Internal	$1399.00
10 meg Fixed/Internal	$1249.00
15 meg 5 Removable/10 Fixed	$2149.00
25 meg 5 Removable/20 Fixed	$2499.00

i-MEGA

10 meg Bernoulli Box	$2149.00
20 meg Bernoulli Box	$2799.00

RanaSystems

10 meg Internal	$899.00

TALLGRASS TECHNOLOGIES

COMMITTED TO MEMORY

12, 25, 35, 50, 80 meg (PC) from $1499.00

FLOPPY

INDUS

Apple GT	$209.00
Atari GT	$249.00
C-64 GT	$259.00

µ-SCI

A1 5 Apple	$199.00
A2 Apple	$199.00

MSD

SD1 C-64 Single	$269.00
SD2 C-64 Dual	$469.00

RanaSystems

Rana 1000 (Atari)	$199.00
Elite 1 (Apple)	$189.00

Tandon

320K 5¼" (PC)	$129.00

MODEMS

ANCHOR

Volksmodem	$59.99
Volksmodem XII	$189.99
Mark II, Serial	$79.99
Mark VII (Auto Ans/Auto Dial)	$99.99
Mark XII (1200 Baud)	$259.00

Hayes

Smartmodem 300	$199.00
Smartmodem 1200	$389.00
Smartmodem 1200B	$359.00
Smartmodem 2400	$699.00
Micromodem IIe	$249.00
Smart Com II	$69.99
Chronograph	$199.00

AST

Reach 1200 Baud Half Card	$399.00

mpp MICROBITS

MPP-1000E AD/AA (Atari)	$109.00
MPP-1064 AD/AA (C-64)	$69.00

Novation

Smart Cat Plus	$289.00
Smart Cat 103	$189.00
Smart Cat 103/212	$369.00
AutoCat	$189.00
212 AutoCat II	$499.00
Apple Cat II	$229.00
212 Apple Cat II	$379.00
Apple Cat 212 Upgrade	$229.00
Macmodem	$319.00

TELELEARNING

C64 300 Baud	$49.99

ZENITH

ZT-1	$339.00
ZT-10	$309.00
ZT-11	$369.00
Z-22 Video Data Terminal	$529.00

DISKETTES

maxell.

5¼" MD-1	$17.99
5¼" MD-2	$23.99
8" FD-1	$39.99
8" FD-2	$49.99

Verbatim.

5¾" SS/DD	$21.99
5¼" DS/DD	$29.99

Bib

5¼" Disk Head Cleaner	$14.99

Dennison

Elephant 5¼" SS/SD	$14.99
Elephant 5¼" SS/DD	$18.99
Elephant 5¼" DS/DD	$19.99
Elephant EMSP 5¼	$24.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-In-File 10	$3.99
Flip-In-File 50	$17.99
Flip-In-File 50 w/lock	$24.99
Flip-In-File (400/800 RDM)	$17.99

AMARAY

50 Disk Tub	$9.99

GRAPHICS

Koala

Atari	$39.99
C 64	$59.99
IBM	$89.99
Apple/Franklin	$79.99

PPI

Super Sketch Pad (C-64)	$39.99
Super Sketch Pad (Atari)	$39.99

Polaroid

Palette	$1399.00

CALL TOLL-FREE

COMPUTER MAIL ORDER

MONITORS

AMDEK
300 Green	$129.00
300 Amber	$139.00
310 Amber IBM-Plug.	$169.00
300 Color/Audio	$239.00
Color 500 Composite/RGB	$389.00
Color 600 Hi-Res (840x240)	$399.00
Color 700 Hi-Res (720x240)	$499.00
Color 710 Long Phosphor	$579.00

BMC
9191U Color	$179.00

NAP
12" Amber	$69.99
12" Amber/Green TTL (ea.)	$119.00

NEC
JB 1260 Green	$99.99
JB 1201 Green	$129.00
JB 1205 Amber	$129.00
JC 1215 Color	$229.00
JC 1218 RGB	$379.00
JC 1460 Color	$269.00
JC 1410 RGB	$669.00

PRINCETON
MAX-12 Amber	$189.00
HX-12 RGB	$469.00
SR-12 RGB	$629.00
Scan Doubler Boerd	$199.00

TAXAN
115 12" Green Mono	$99.99
116 12" Amber Mono	$99.99
121 Green TTL	$139.00
122 Amber TTL	$149.00
210 Color RGB	$239.00
400 Med-Res RGB	$299.00
410 Hi-Res RGB	$339.00
420 Hi-Res RGB (IBM)	$429.00
440 Ultra Hi-Res RGB	$589.00

QUADRAM
8400 Quadchrome	$479.00
8410 Quadchrome II	$489.00
8420 Amberchrome	$179.00

ZENITH
ZVM 122 Amber	$89.99
ZVM 123 Green	$89.99
ZVM 124 IBM Amber	$149.00
ZVM 131 Color	$299.00
ZVM 133 RGB	$429.00
ZVM 135 RGB/Color	$459.00
ZVM 136 RGB/Color	$599.00

INTERFACES

PRACTICAL PERIPHERALS
Graphcard	$79.99
Serial Card	$99.99
Microbuffer II+	$169.00
Microbuffer 32K	$189.00

QUADRAM
Microfazer	from $139.00
Efazer (Epson)	from $79.99

Orange Micro
Grappler CD (C64)	$99.99
Grappler + (Apple)	$89.99
Grappler 16K+ (Apple)	$189.00

DIGITAL DEVICES
Ape Face (Atari)	$49.99
Uprint (Atari)	$59.99
Uprint (C-64)	$59.99
Printer Buffer P-16	$79.99

MPP MICROBITS
MB1150 Parallel (Atari)	$79.99
MPP-1150 Parallel (Atari)	$69.99
Mi-116cTL (Atari 130cTL)	$69.99
MicroStuffer 64K Print Buffer	$109.00

PRINTERS

ATARI
AT-100 Atari Interface Printer	$159.00
AT-550 Atari Dual Mode	$259.00
GP-100 Parallel Interface	$189.00
GP-700 Atari Color Printer	$489.00
GP-550 Parallel Printer	$289.00

BMC
BX-80 Dot Matrix	$229.00

CITIZEN
MSP-10 (80 col)	$349.00
MSP-15 (132 col)	$539.00
MSP-20 (80 col)	$569.00
MSP-25 (132 col)	$729.00

C.ITOH
Prowriter 7500	$219.00
Prowriter 8510P	$299.00
Prowriter 1550P	$469.00
Son of Starwriter A10P	$459.00
Hot Dot Matrix	$459.00
F10-40P Starwriter	$869.00
F10-55 Printmaster	$1049.00

COMREX
ComWriterII Letter Quality	$399.00

DIABLO
620 Letter Quality	$749.00
630 API Letter Quality	$1549.00

daisywriter
2000	$849.00

EPSON
RX-80, FX-80+, LX-80, JX-80	CALL
FX-100+, RX-100, LQ1500	CALL
Homewriter 10	CALL

JUKI
8100	$399.00
8300	$749.00

MANNESMANN TALLY
Spirit 80	$229.00
160L	$499.00

NEC
8027 Transportable	$319.00
2010/15/30/50	$899.00
3510/15/30	$1299.00
3550 IBM	$1369.00
8810/15/30	$1679.00
8850 IBM	$1699.00
ELF 360/370	CALL

OKIDATA
82, 83, 84, 92, 93, 182, 2350, 2410, Qkimate-20	CALL
Qkimate 10 (Specify C64/Atari)	$199.00

OLYMPIA
Needlepoint Dot Matrix	$329.00
Compact RO	$399.00
Compact 2	$399.00

Panasonic
KX1090	$199.00
KX1091	$279.00
KX1092	$409.00
KX1093	$599.00

QUADRAM
Quadjet	$749.00

SILVER-REED
400 Letter Quality	$279.00
500 Letter Quality	$329.00
550 Letter Quality	$429.00
770 Letter Quality	$779.00

star
SG10 (120 cps)	$239.00
SG15 (120 cps)	$399.00
SD10 (160 cps)	$359.00
SD15 (160 cps)	$479.00
SR10 (200 cps)	$499.00
SR15 (200 cps)	$639.00
Powertype Letter Quality	$319.00

TOSHIBA
1340 (80 column)	$599.00
1351 (132 column)	$1259.00

PC COMPATIBLES

ZENITH
PC-150 Desktop	CALL
PC-160 Portable	CALL

COLUMBIA DATA PRODUCTS INC
2220 Duel Portable	$1999.00
4220 Duel Desktop	$1999.00

SANYO
MBC 550-2 Single Drive	$699.00
MBC 555-2 Dual Drive	$969.00
MBC 775 Portable	CALL

AT&T
Satari	CALL
6300	CALL

corona
PPC22 Duel Portable	$1599.00
PPCXTA 10 meg Portable	$2799.00
PC40022 Dual Desktop	$2199.00

NEC
PC-8600 8/16 Bit	$1299.00

SOFTWARE FOR IBM

Lotus
Symphony	$429.00
1-2-3	$299.00

Hayes
Please (Data Base)	$199.00

Peachtree Software
PeachPack (GL/AP/AR)	$199.00

MicroPro
WordStar 2000	$239.00
WordStar 2000+	$299.00

MICRORIM
R Base 4000	$249.00
Clout 2.0	$129.00

MultiMate
Multi Mate	$249.00

MICROSTUF
Crosstalk	$89.99

MICROSOFT
Flight Simulator	$39.99
MultiPlan	$129.00

ASHTON-TATE
Framework	$349.00
dBASE II	$299.00
dBASE III	$369.00

Professional Software
PC Plus/The Boss	$249.00

synapse
File Manager	$49.99

ELECTRONIC ARTS
Get Organized	$69.99
Cut -n- Paste	$39.99
Music Construction	$29.99
One -on- One	$29.99
Financial Cookbook	$34.99

ALPHA
Electronic Desk	$199.00

BORLAND INTERNATIONAL
Turbo Pascal	$39.99
Sidekick	$39.99

SPI
Open Access	$379.00

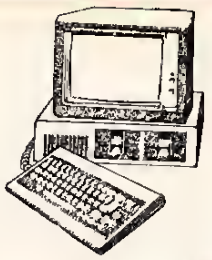
Harvard Software Inc.
Harvard Project Manager	$209.00
Total Project Manager	$269.00

pfs:
IBM/APPLE
Access (NEW)	$79.99
Write/Graph/File (ea.)	$79.99
Report	$79.99
Plan	$79.99
Proof	$59.99
Mac Software	CALL

SSI
Word Perfect	$239.00

Human Edge™
Communication Edge	$99.99
Management Edge	$119.00
Negotiation Edge	$139.00
Sales Edge	$119.00

IBM



IBM PC SYSTEMS
Configured to your specifications.
Call for Best Price!

MULTIFUNCTION CARDS

AMDEK
MAI	$249.00

AST
Six Pack Plus	$239.00
Mega Plus II	$269.00
I/O Plus II	$139.00
Memory MBII	$249.00
Advantage-AT	$399.00
Preview Monograph	$299.00
Graph Pak Mono/64K	$599.00
MonoGraph Plus	$399.00
5251/11	CALL
5251/12	$579.00
3780	$639.00
BSC	$499.00

dca
IRMA 3270	$899.00
IRMA Print	$999.00

EAGLE
Color	$199.00
Mono	$149.00

EVEREX
Color Card (Graphics Edge)	$299.00
Magic Card	$199.00

HERCULES
Graphics	$319.00
Color	$169.00

IDEAssociates
IDEAmax - ZPR, 64K, C, S, P	$229.00
IDEAmini - YPR, C, S, P	$189.00
IDEAminimax - MPR 128K	$229.00
IDEAshare Software	$219.00
IDEA 5251	$699.00

PARADISE
Modular Graphics Card	$279.00
Multi Display Card	$299.00
Five Pack C, S	$159.00

PLANTRONICS
Color Plus	$369.00

TECMAR
Captain - 64	$239.00
Captain Jr 128K	$339.00
Graphics Master	$469.00

QUADRAM
Quadboard II	$229.00
Expanded Quadboard	$239.00
Quad 512+	$249.00
Quad 2 Meg	$879.00
Memory Board	$129.00
QuadJr Exp. Chassis	$539.00
QuadJr Exp. Memory	$219.00
QuadMemJr	$229.00
Chronograph	$89.99
Parallel Card	$69.99
Quadcolor I	$219.00
Quadgraph	$379.00

ACCESSORIES

KEYBOARDS
Keytronics Keyboards 5150, 5151, 5151 Jr, 5149 Jr	CALL

MEMORY CHIPS
4164 RAM Chips (ea.)	$2.39

1-800-233-8950

SpeedScript, you send a control code to the printer which backspaces and underlines after each character. But to Commodore 1525 or 801 printers, the code that most other printers understand as backspace is read as a command to enter graphics mode. Such conflicts are unavoidable, because there are so many different printers and control codes.

Whether you're just getting started with printers or are moving on to advanced printing features, there are a few basic concepts you should understand. If you're having printer problems, check this list to be sure you're familiar with each item. If you're not, invest some time exploring your computer, printer, software, and interface manuals to find a solution.

• **ASCII** (pronounced "AS-key"). American Standard Code for Information Interchange. A code that uses numbers from 0 to 127 to represent letters, numbers, punctuation symbols, and special control codes. Each code number consists of seven *bits* (binary digits). An eighth bit may be added for *parity* (see below). The first 32 ASCII numbers are control codes which can tell your printer to perform actions such as linefeeds, carriage returns, backspaces, and vertical and horizontal tabs.

Commodore and Atari computers use a slightly different form of ASCII which can cause translation problems with some interfaces and printers unless they are correctly configured.

• **Baud rate**. A measure of data transmission speeds, synonymous at lower speeds with bits per second. Computers can send data much faster than printers can produce images on paper. Consequently, the printer and interface must tell the computer to send data in bursts short enough for the printer to handle.

• **Buffer**. In a printer or interface, the memory area in which data is held after being sent from the computer. Printer buffers can be as small as one line of characters or range upward to thousands of bytes of data. If the buffer is large enough, it can hold all of the data you want to print, thus freeing the computer for other tasks while the printer goes about its work. The printer controls the speed at which the data leaves the buffer and is printed on the paper.

• **Centronics-standard parallel connection**. A printer interface which allows data to be sent along separate wires eight bits at a time in a parallel flow. Most printers use a parallel interface to receive data from the computer. Some computers, however, must transmit data through a *serial* interface (see *RS-232-standard serial connection*). The Centronics interface, named after the printer company which popularized it, is the most common type of parallel interface on personal computers.

• **Character set**. The letters, numbers, and symbols which a printer or computer can produce. Note that many computers can display characters which the printer cannot reproduce, and vice versa. Some printers are capable of printing foreign character sets when you change the DIP switch settings. (See *DIP switches*).

• **Control codes**. Nonprintable commands sent from the computer to a printer for special actions, such as backspacing, carriage returns, linefeeds, tabs, and margin settings.

• **DIP switches** (Dual In-line Package). Small switches located on a printer or interface which can control a variety of options, such as baud rates, automatic/manual linefeeds, printing impression levels, international character sets, types of paper, form lengths, line spacing, and other parameters. Some printers and interfaces allow easy access to DIP switches, while others require you to take apart the case.

• **Emulation**. In terms of printers, a mode of operation which mimics another type of operation. For instance, some printer interfaces let a non-Commodore printer emulate a Commodore printer, allowing you to print the computer's special graphics symbols and reverse-video characters.

• **Escape codes**. Control code sequences which let you print certain characters not included in ASCII codes, or which activate special printer features such as boldfacing, italics, expanded or condensed type, and so on. Escape sequences are preceded by the *escape character*, ASCII 27. These sequences can be sent to the printer in BASIC from your computer keyboard, or by the application software (such as a word processor). For example, ESC H might represent a British pound sign, ESC P might turn on or off the proportional spacing option on your printer, and ESC BS may determine the amount of space between characters in backspacing.

• **Firmware**. Software permanently burned into a ROM chip (Read Only Memory)—a cross between hardware and software. Printers contain firmware to control their printing options. Sometimes you can replace this chip with another to add more printing features.

• **Parity**. A way for your computer and printer to check the accuracy of the data being sent. An extra bit is added to the end of a seven-bit ASCII code representing a particular character. The computer checks the extra bit to verify that the data was not scrambled during transmission.

• **Proportional spacing**. Many printers today can vary the spacing between characters, as typesetters do. Typewriters have fixed spacing between all letters. For example, proportional spacing allows more room for a capital M or W and less room for a lowercase i or l.

• **Protocol**. All the rules and instructions controlling the way in which data is sent and received between the computer and printer.

• **RS-232-standard serial connection**. A type of interface that transmits data along a single wire one bit at a time, or serially. Although most printers use a parallel interface to receive data from a computer, some computers and printers require a serial interface. When all other factors are equal, a serial interface is slower than a parallel interface—but this is rarely important with printers, whose speeds are determined by mechanical limitations anyway.

• **Tractor-feed**. A pair of cogged wheels and guide wires that helps continuous-form paper roll through a printer. Some computers have built-in tractors, and others offer them as options.

• **Transparency**. A mode of operation for printer interfaces in which serial data is changed to parallel data without converting the original values of the data. ©

COMPUTE! BOOKS'

Kids and the IBM PC

Edward H. Carlson

If you are acquainted with BASIC, you can easily write your own games and applications. Thirty-three sections are included, with instructor notes, lessons, assignments, and lively illustrations to entertain and amuse you. *Kids and the IBM PC and PCjr* is also a good tutor on programming your PC or PCjr that will offer the best possible introduction to BASIC.

Edited

Now for the Commodore 64 and the Commodore 128, this collection brings together some of the best games, applications, and utilities from COMPUTE! Publications. All programs run on the 64 and the 128 running in 64-mode. Additionally, there are sections detailing the advanced special features of the powerful, new 128 computer. *Disk available which includes programs in the book.*

Thomas E. Enright, Joan Nickerson, and Anne Wayman

This informative and easy-to-understand book shows the beginner how to use the Apple IIc to communicate with other computers over the phone. Bulletin boards and informational services such as The Source, CompuServe, and Dow Jones are fully explored. Other useful information describes, in plain English, everything a buyer needs to know before selecting a modem or telecommunications software for the IIc.

Charles Brannon

SpeedScript, the most popular program ever published by COMPUTE! Publications, is a commercial-quality word processor for the expanded VIC and Commodore 64 computers. Included are all the programs and documentation necessary for both versions of *SpeedScript*. In addition, we've included source code and documentation about *SpeedScript* that have never been published before. For the price of the book, you get a commercial-quality word processor—perhaps one of the best software bargains ever. *Disk available which includes programs in the book.*

Arlan R. Levitan and Sheldon Leemon

Learn the ins and outs of telecomputing on your IBM PC or PCjr. From selecting a modem to evaluating terminal software, *COMPUTE!'s Telecomputing on the IBM* will guide you through the steps in clear, everyday language. Getting online with a local bulletin board or an information service such as Dow Jones, The Source, or CompuServe is made easy by the explanations offered in this book. There's a world of information available to you through your IBM, and *COMPUTE!'s Telecomputing on the IBM* will help get you online quickly and easily.

Don Gutman and Shoy Addams

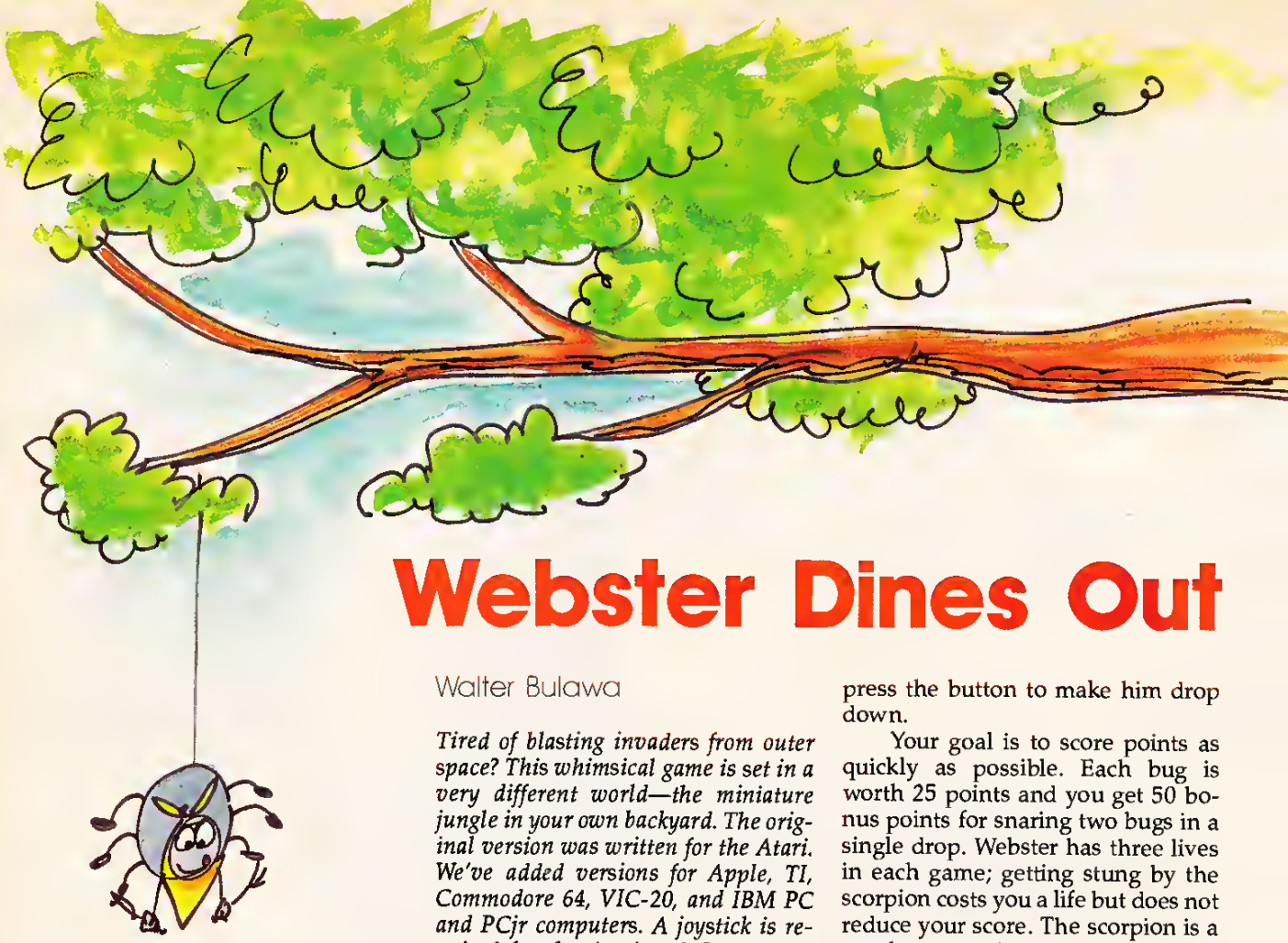
With hundreds of computer games available for every home computer, how's a game enthusiast to decide which ones to buy? *The Greatest Games*, written by the founders and editors of *Computer Games* magazine, has the answers. It contains lively, in-depth reviews of the 93 greatest computer games. Eighteen different types of games are reviewed, including role-playing games, graphic adventures, sports games, and classic arcade games. This is a book every computer game lover will want to own.

Christopher Flynn

A complete home applications system of 25 integrated programs that truly put the speed and power of the PC or PCjr to work in the home. The system includes a spreadsheet, appointment calendar, electronic filing system, graph creation programs, and much more. Each program is clearly explained, well-documented, and will run on the PC, with or without the Color/Graphics Monitor Adapter, as well as on any PCjr. *Disk available which includes programs in the book.*

These titles are available at your local book or computer store, or you may order directly from COMPUTE! Books.

To order, call toll free 800-334-0868, or send your check or money order (including $2.00 per book for shipping and handling) to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.



Webster Dines Out

Walter Bulawa

Tired of blasting invaders from outer space? This whimsical game is set in a very different world—the miniature jungle in your own backyard. The original version was written for the Atari. We've added versions for Apple, TI, Commodore 64, VIC-20, and IBM PC and PCjr computers. A joystick is required for the Atari and Commodore 64 versions.

Guide Webster, the hungry tree spider, in his endless search for a square meal. Roving back and forth across his tree limb, he watches for bugs to appear in the grass below. When the time is right, he drops down on a strand of silk for a light snack, then climbs back up his web to look for more.

Unfortunately, this backyard paradise isn't quite perfect. The more Webster eats, the faster the bugs move, making it harder to find the next meal. Even worse, he's not the only one with an appetite—there's a speedy scorpion sharing the same hunting ground, stealing bugs when he can and giving Webster a sting whenever he drops too close.

Atari Version

Program 1—the Atari version of "Webster Dines Out"—will run on any Atari computer with at least 32K memory. Use the joystick to move Webster left or right at the top of the screen. When a bug passes below, press the button to make him drop down.

Your goal is to score points as quickly as possible. Each bug is worth 25 points and you get 50 bonus points for snaring two bugs in a single drop. Webster has three lives in each game; getting stung by the scorpion costs you a life but does not reduce your score. The scorpion is a tough competitor: When Webster drops down, the scorpion speeds up to increase his chances of stealing a bug.

There are six skill levels, each harder than the last. As you advance to higher levels, the bugs and scorpion speed up, the grass grows longer, and a grey rock appears in the lawn. The other creatures hide behind these objects, but Webster can drop behind them too. The game ends when you lose all three lives or exhaust your time at the highest skill level.

Commodore 64 And VIC-20 Versions

Both Commodore versions of Webster Dines Out are scored like the Atari game—25 points for each bug, with a 50 point bonus for capturing two at once. You begin with three lives, and lose one each time you collide with the scorpion.

The 64 version (Program 2) is played with a joystick in port 2. The bugs and scorpion move across a sloping, multicolored lawn; at higher skill levels, colorful objects grow up to obscure your view of the ground. Play ends when your lives

Patterson Blair

Now Available For COMMODORE

Looks like a Ferrari. Drives like a Rolls. Parks like a Beetle.

GT for Atari shown

Ask your computer dealer to let you test drive the all new Indus GT.™

The most advanced, most handsome disk drive in the world.

Flip its power switch and...
Turn your Atari into Ferrari.
Unleash your Apple.
And now turbocharge your Commodore.

Looks like a Ferrari.

The Indus GT is only 2.65" high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Engaging its AccuTouch™ buttons lets you control the LED-lit CommandPost.™ Marvel at how responsive it makes every Commodore, Apple and Atari personal computer.

Drives like a Rolls.

Nestled into its soundproofed chassis is the quietest and most powerful disk drive system money can buy. At top speed, it's virtually inaudible...whisper quiet.

Built into each Indus GT is a perfect combination of craftsmanship and advanced engineering. Luxurious styling reflects the personal tastes of each GT owner.

And each GT comes with the exclusive *GT DrivingSystem*™ of software programs.* World-class word processing is a breeze with the *GT Estate WordProcessor*.™ Your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills.

Also, the Indus GT is covered with the GT PortaCase.™ A stylish case that conveniently doubles as a 80-disk storage file.*

Parks like a Beetle.

The GT's small, sleek, condensed size makes it easy to park.

A WarrantyPlus™ package is included with every Indus GT, featuring full year parts and labor on the complete drive train.

Drive home a winner and park an Indus GT next to your personal computer.

INDUS™

The all-new Indus GT Disk Drive.

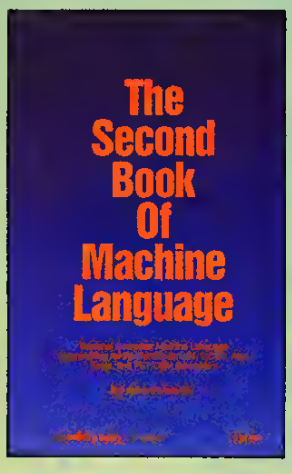
The most advanced, most handsome disk drive in the world.

*Included as standard equipment

For dealer information, call (818) 882-9600

BUY THE BEST—BEST-SELLING TITLES FROM **COMPUTE! BOOKS**

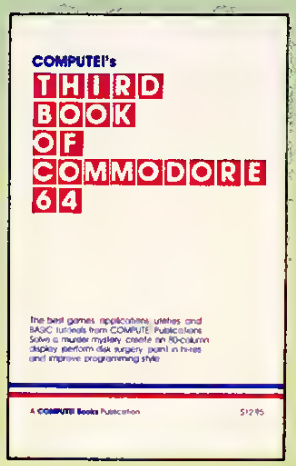
We offer you six best-selling books designed especially for your home, educational, and recreational computing. Each book is filled with step-by-step instructions, helpful tutorials, and creative hands-on applications. COMPUTE! books are carefully written so that any computer user can easily understand and enjoy them to the fullest. This industry-leading selection includes:



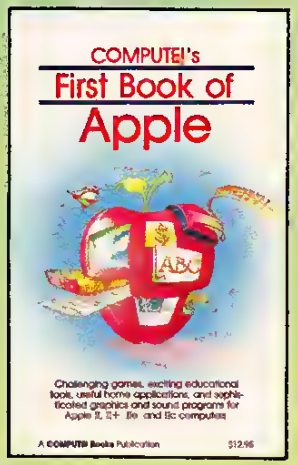
The Second Book of Machine Language
Richard Mansfield
$14.95
The follow-up to the best-selling *Machine Language for Beginners*, this book shows how to construct significant, effective machine language programs. It includes a high-speed, professional-quality, label-based assembler, and everything that's needed for optimized programming on the Commodore 64, Apple, Atari, VIC-20, and PET/CBM computers.

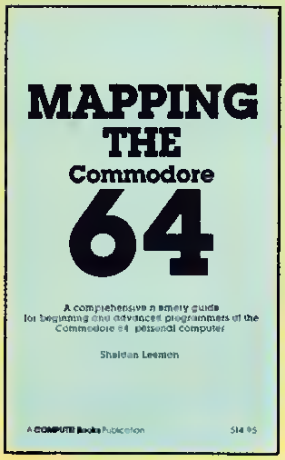
COMPUTE!'s Commodore Collection, Volume 2
Edited
$12.95
This second volume in COMPUTE!'s Commodore Collection series includes exciting games, sophisticated applications, versatile educational routines, and helpful programming aids for VIC-20 and Commodore 64 owners.

COMPUTE!'s Third Book of Commodore 64
Edited
$12.95
The third in a collection of informative, entertaining, and educational games, applications, tutorials, and utilities for Commodore 64 users.



COMPUTE!'s First Book of Apple
Edited
$12.95
A collection of 35 exciting games, educational programs, home applications, and graphics routines for owners of Apple II, IIe, and IIc computers.





Mapping the Commodore 64
Sheldon Leemon
$14.95
An excellent and comprehensive memory map and programming guide for intermediate to advanced programmers.

COMPUTE!'s Machine Language Routines for the Commodore 64
Edited
$14.95
Complete machine language programs and easy-to-use routines you can learn from and use in your own programming.

You'll find these and other best-selling titles from COMPUTE! Books at a store near you listed in the following pages. Visit one of these stores today to find COMPUTE! books, the best in personal computer publications.

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies
324 W Wendover Avenue Suite 200 Greensboro, NC 27408 919-275-9809
Publishers of COMPUTE! COMPUTE!'s Gazette COMPUTE!'s Gazette Disk COMPUTE! Books and COMPUTE!'s Apple Applications

Look for **COMPUTE!** Books at the following stores.

Alabama

Madison Books and Computers
Madison Mall
8006-S Madison Pike
Madison, AL 35758

Arizona

Books Etc
901 S Mill Ave
Tempe, AZ 85281

California

Aztec Shops, Ltd
San Diego State University
San Diego, CA 92182

Associates Students UCLA
General Book Dept
308 Westwood Plaza
Los Angeles, CA 90024

Bookmania
25590 Seaboard Lane
Hayward, CA 94545

Computer Literacy
520 Lawrence Expressway #310
Sunnyvale, CA 94086

M T S Inc
Tower Books Div
7830 Macy Plaza Dr
Citrus Heights, CA 95610

M T S Inc
Tower Books #14
253B Watt Ave
Sacta, CA 95821

University Bookstore, Q008
University of California, San Diego
La Jolla, CA 92093
(619) 452-3149

University Bookstore
University Avenue East
P.O. Box 5800
Riverside, CA 92507

University Bookstore
University Center
University of California
Irvine, CA 92717
(714) 856-7412

Palomar College Bookstore
1140 W Missian
San Marcos, CA 92069

Titan Bookstore
4999 Casa Lama Ave
Yorba Linda, CA 92686

Connecticut

Klein's of Westport
44-50 Main Street
Westport, CT 06880

UConn Co-op
61 Fairfield Road
Storrs, CT 06268

Delaware

Castle Video, Inc
27 Chestnut Hill Plaza
Newark, DE 19713

McMahon's
101 Christiana Mall
Newark, DE 19702

Encore #8122
Talleyville SC
Silverside Rd. Rt 202
Wilmington, DE 19810

District of Columbia

Reiter's Scientific and Professional Books
2120 Pennsylvania Ave N W
Washington, DC 20037

Florida

Paperback Booksmith
400 Northlake Blvd
North Palm Beach, FL 33408

DuBeys News Center
115 S Monroe Street
Tallahassee, FL 32301

The Book Nook
4345-2 University Blvd S
Jacksonville, FL 32216

Book Trek
3734 Blanding Blvd
Jacksonville FL 32210

Monocle Book Store
Tamp Int A P
Tampa, FL 33607

Georgia

Oxford Books Inc
2345 Peachtree Rd., N E
Atlanta, GA 30305

Hawaii

Honolulu Book Shops, Ltd
287 Kalihi Street
Honolulu, HI 96819

Illinois

Krochs & Brentanos
1028 Lake St
Oak Park, IL 60301

Krochs & Brentanos
J-6 Evergreen Plaza
Evergreen Park, IL 60642

Krochs & Brentanos
18 River Oaks Center
Calumet City, IL 60409

Krochs & Brentanos
159 Lincoln Mall
Matteson, IL 60443

Krochs & Brentanos
E-108 Cherry Vale Mall
Rockford, IL 61112

Krochs & Brentanos
Randhurst S/C
999 Elmhurst Rd
Mt Prospect, IL 60056

Krochs & Brentanos
2284 Fox Valley Center
Aurora, IL 60505

Krochs & Brentanos
100 Orland Square
Orland Park, IL 60462

Krochs & Brentanos
1711 Sherman Ave
Evanston, IL 60201

Krochs & Brentanos
36 Old Orchard Ctr, N Mall
Skokie, IL 60077

Krochs & Brentanos
80 Oakbrook Ctr
Oakbrook, IL 60521

Krochs & Brentanos
106 Hawthorne Ctr
Vernon Hills IL 60061

Krochs & Brentanos
J-108 Woodfield Mall
Schaumburg, IL 60195

Krochs & Brentanos
516 N Michigan Ave
Chicago, IL 60611

Krochs & Brentanos
835 N Michigan Ave
45th Floor Water Tower Plaza
Chicago, IL 60611

Krochs & Brentanos
105 W Jackson
Chicago, IL 60604

Krochs & Brentanos
29 S Wabash
Chicago, IL 60603

Indiana

Von's Bookshop Computer Center
315 W State St
W Lafayette, IN 47906

Book World
192 University Park Mall
Mishawaka, IN 46545

Louisiana

SoundTrek Audio
5703 Read Blvd
New Orleans, LA 70127

SoundTrek Audio
4420 I-10 Service Rd
Metane, LA 70001

SoundTrek Audio
400 Lapalco Blvd
Gretna, LA 77053

Maryland

R W Electronics & Computer Ctr
8070 A Jumpers Mall
Pasadena, MD 21122

Collins Brothers Enterprises
Computer Corner
29 N Potomac St
Hagerstown, MD 21740

Rockville Sales
11807 Tidewood Rd
Silver Spring MD 20906

Software Store
842-A Rockville Pike
Rockville, MD 20852

Software Store
7315 Wisconsin Ave
Bethesda, MD 20814

Maryland Book Exchange
4500 College Ave
College Park, MD 20740

Massachusetts

Boston Univ Bookstore
660 Beacon Street
Boston, MA 02215

Northeastern University Bookstore
360 Huntington Avenue
Boston, MA 02115

Michigan

Chi Town Records
228 State St
Oscada, MI 48750

I O Software
3600 South Dort Hwy
Flint, MI 48507

Little Professor Book Center
6387 W Grand River
Brighton, MI 48116

Little Professor Book Center
22174 Michigan Avenue
Dearborn, MI 48124

Little Professor Book Center
37115 Grand River
Farmington, MI 48024

Merit Books
22425 Kelly Road
East Detroit, MI 48021

Metro News
6608 Telegraph
Birmingham, MI 48010

Town Center Bookstore
3000 Town Center
Southfield MI 48075

New Hampshire

Marjen Books, Inc., dba
Paperback Booksmith/Musicsmith
Newington Mall
Portsmouth, NH 03801

The Colonial Bookshoppe
Upper Valley Plaza
West Lebanon, NH 03784

New Jersey

Village Computer & Software Center
Morris County Mall
Ridgedale & Hanover Aves
Cedar Knolls, NJ 07927

Encore Books #8104
The Market Place SC
1990 Marlton Pike
Cherry Hill, NJ 08003

Encore Books #8117
Eagle Plaza SC
34 Berlin Rd
Voorhees, NJ 08043

Encore #8121
Village at Taunton Forge
Tuckerton Tauton Rds
Medford, NJ 08055

The Princeton University Store
36 University Place
Princeton, NJ 08540

The Program Store
Rt 35 & Wycoff Road
Monmouth Mall
Eatontown, NJ 07724

The Program Store
1344 Willowbrook Mall
Wayne, NJ 07470

New York

Software City
134 Village Landing
Fairport, NY 14450

Village Green Bookstore
766 Monroe Ave
Rochester, NY 14607

McGraw Hill Bookstore
135 W 48th St
New York, NY 10020

Book Revue
313 New York Ave
Huntington, NY 11743

Macy's
Book Dept
Herold Square
New York NY 10001

North Carolina

The Book Collection Inc
117 North Greene St
Greensboro, NC 27410

News & Navels
West Market St Center
Greensboro, NC 27407

Wills Books
Friendly Shopping Center
Greensboro, NC 27401

Wills Books
Four Seasons Mall
Greensboro, NC 27403

Ohio

Wit & Wisdom, Inc
dba Booksellers
24031 Chagrin Blvd
Beachwood, OH 44122

Barnes & Noble Bookstore, Inc
dba Cleveland State Univ
2400 Euclid Avenue
Cleveland, OH 44115

The Inside Story
8535 Tanglewood Mall
Chagrin Falls, OH 44022

Bookstore at the University of Akron
Barnes & Noble
303 E Carrall St
Akron, Ohio 44325

Cuyahoga Community College
Western Campus Book Center
11000 Pleasant Valley Road
Parma, OH 44130

University Bookstore
Bowling Green State University
Bowling Green, OH 43403

Oregon

Book Vault
Bernard's Mall
3125 S W Cedar Hills Blvd
Beavertan, OR 97005

Pennsylvania

Computer Corner
Bazaar of all Nations
Store #201 Baltimore Park
Cliftan Hts, PA 19018

Encore Books #8101
1413 Walnut St
Philadelphia, PA 19102

Encore Books #8103
205 South 38th St
Penn Campus
Philadelphia, PA 19104

Encore Books #8107
609 Chestnut St
Philadelphia, PA 19106

Encore Books #8112
34 South 17th St
United Engineers Bldg
Philadelphia, PA 19103

Encore Books #8115
Chestnut Hill/Top of Hill
55 Bethlehem Pike
Philadelphia, PA 19118

Encore Books #8118
Cottman Bustleton SC
2119 Cottman Ave
Philadelphia, PA 19149

Encore Books #8119
Tremont Shopping Center
2536-38 Welsh Road
Philadelphia, PA 19152

Encore Books #8105
205 West Lancaster
Wayne, PA 19087

Encore Books #8106
457 Germantown Pike
Plymouth Meeting, PA 19462

Encore Books #8108
23 Manoo Shopping Center
Eagle Rd West Chester Pk
Havertown, PA 19083

Encore Books #8109
Ardmore West SC
139 W Lancaster Ave
Ardmore, PA 19003

Encore Books #8115
Media Shipping Center
519 Baltimare Pike
Media, PA 19063

Encore Books #110
455 Old York Rd
Jenkintown, PA 19046

Encore Books #8111
Lawrence Park SC
Rt 302 at Lowrence Rd
Broomall, PA 19008

Encore Books #8116
Hampton Sq Shop Ctr
Second Street Pike
Southampton, PA 18966

Encore Books #8120
Gateway Shopping Ctr
Rt 202 S Valley Forge Rd,
Wayne, PA 19087

South Carolina

Happy Bookseller
34 Richland Mall
Columbia, SC 29204

Texas

Taylors #1
5455 Beltline Rd
Dallas, TX 75240

Taylors #2
4001 Northwest Pkwy
Dallas, TX 75225

Taylors #3
254 Lincoln Square
Shopping Center
Arlington, TX 76011

Taylors #4
6346 Camp Bowie Blvd
Ft Worth, TX 76116

Fort Worth Book & Video
400 Main
Fort Worth, TX 76102

Videoland
2131 N Town East Blvd
Mesquite, TX 75150

Videoland
6827 Green Oaks Road
Ft Worth, TX 76116

Videoland
4750 FM 1960
Houston, TX 77069

Videoland
7080 SW Fwy
Houston, TX 77074

Babbages
Town & Country Mall
800 W Belt
Houstan, TX 77024

Babbages
7925 West FM 1960
1536 Willowbrook Mall
Houston, TX 77070

Babbages
Baybrook Square
1369 West Bay Area Blvd
Houston, TX 77598

Babbages
170 Greenspoint Mall
Houston, TX 77060

Babbages
1000 West Oaks Mall
Houston, TX 77082

Videoland
13340 Preston Rd
Dallas, TX 75240

Babbages
Northpark Mall
455 NorthPark Center
Dallas, TX 75225

Babbages
2228 Valley View Mall
Dallas, TX 75240

Federated/Superstore #3240
12002 S W Freeway
Meadows, TX 77477

Federated/Superstore #2110
19801 Gulf Freeway
Webster, TX 77948

Federated/Superstore #1101
100 Cypress Wood
Spring, TX 77388

Babbages
1130 Collin Creek Mall
811 North Centrol Expressway
Plano, TX 75075

Babbages
2060 Town East Mall
Mesquite, TX 75150

Babbages
2064 Northeast Mall
Hurst, TX 76053

Babbages
243 Hulen Mall
4800 South Hulen Street
Ft Worth, TX 76132

Babbages
3843 Irving Mall
Irving, TX 75062

Virginia

Crest Books Towncenter
115 Harry Flood Byrd Highway
Sterling, VA 22170

Wards Corner Book Shop, Inc
7524 Granby Street
Norfolk VA 23505

Software Store (Vienna)
8601 Westwood Ctr Dr
Vienna, VA 22180

Washington

The Programs Plus
16850 S Center Pkwy
Seattle, WA 98188

are used up or time runs out at the last skill level.

The VIC-20 version of Webster Dines Out is written in machine language (ML) for the unexpanded VIC. Program 3 is a BASIC loader that saves the ML program on disk or tape. Since the loader won't fit in an unexpanded VIC, you'll need at least 8K memory expansion to run it (a Commodore 64 can also be used; see instructions below). Type in and save Program 3, but don't try to run it yet. Enter this line in direct mode (without a line number):

```
POKE6609,0:POKE43,209:POKE44,25:
NEW
```

Now reload Program 3 and run it. Press D to save the game on disk, or T to save it on tape. The finished program will be named WEBSTER (replacing any other program of that name on your disk). If you don't have memory expansion, you can use a Commodore 64 to create the VIC game (of course, the game itself runs only on a VIC). To run the loader on a 64, change the 57809 to 57812, and 63109 to 62957 in lines 9–11 of Program 3. Then follow the procedure described above.

Once the game is saved, remove any memory expansion and load and run it like a BASIC program. Move Webster with the < and > keys, and drop him down by pressing the space bar. As you progress to higher skill levels, the speed increases, and rocks appear below, blocking your vision. You can snare bugs from behind the rocks, but be careful not to drop onto a hidden scorpion. Play continues until you lose all three lives.

IBM Version

Program 4 runs on any PCjr with cartridge BASIC and any IBM PC with BASICA and a color/graphics adapter card. Press the left and right cursor keys to move Webster, and the space bar to drop.

Webster's lifeline is displayed at the top of the screen. When you drop to get a bug, your energy level is drained and your lifeline shrinks. Capturing a bug restores your energy and expands your lifeline. You'll score 10 points for catching a beetle, and 20 for each bug, with bonus points for multiple captures. Extra bonus points are awarded at the 1,000, 5,000 and 10,000 point marks. As your score increases, the bugs speed up and become more scarce; your energy will drain faster, too. The game ends when you hit the scorpion or your energy drains to zero.

Apple Version

This version of Webster Dines Out will run on any Apple II series computer. Since it's written entirely in machine language (ML), it must be entered using the "Apple MLX" machine language editor found elsewhere in this issue. MLX will greatly simplify the usually tedious job of accurately entering the many numbers that make up a ML program. But be sure that you read the MLX article and understand how to use MLX before you begin entering the data from Program 5.

When you run MLX, it will ask for a starting and ending address. Use the values indicated in Program 5:

```
START ADDRESS?    1100
END ADDRESS?      1F14
```

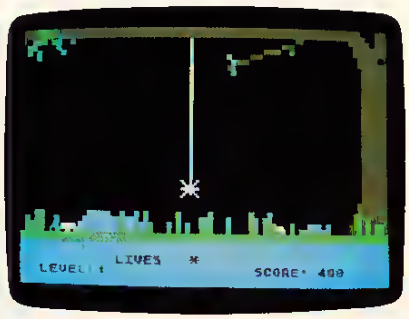
MLX will then give you a menu of options. Choose E for enter and give 1100 as the starting address. A prompt for the first line will appear, and you can begin entering the data from Program 5. If you don't type the entire listing in one sitting, follow the instructions in the MLX article for saving a partially complete version and reloading it later. When you're finished typing, MLX will prompt you for a filename for the completed machine language program. To load and run the game, simply type BRUN"WEBSTER" (or whatever name you used for the completed program) and press RETURN.

The scoring is identical to the Atari version. Use the left and right arrow keys to move on the branch, and press the space bar to drop to the ground. Avoid colliding with the giant grasshopper—when that happens, Webster loses a life (and is carried bodily off the screen). The grass in the lawn grows higher as the game progresses, making your job more difficult. You can drop into the grass to snare a hidden bug, but be sure to keep track of the giant hopper, who might be lurking there as well.

TI Version

This version of Webster (Program 6) uses sprites for the spider, bugs, and scorpion, and thus requires TI Extended BASIC. You can use either joystick or keyboard controls. For the keyboard, press the S key to move left, the D key to move right, and the space bar to make Webster drop. Scoring is identical to the Atari version: You have three lives in each game, and six possible levels. Bugs are worth 25 points, with a 50 point bonus for catching two at once.

At the bottom of the screen you'll see Webster's lifeline. Dropping for a bug drains your energy and shrinks the lifeline; catching a bug restores your energy to normal. You lose a life whenever your energy drains to zero or you hit the scorpion. The game will not end until you lose all three lives.



Growing grass and grey rocks make it tough for Webster to find a meal in Atari "Webster Dines Out."

Program 1: Webster Dines Out For Atari

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
KI 10 GOSUB 7000
KK 20 GOSUB 7100
LA 30 GOSUB 7600
OO 50 LEVEL=LEVEL+1
CL 80 ON LEVEL GOSUB 85,1000
      0,10100,10200,10300,10
      400,19000
PL 85 POKE 656,2:POKE 657,8:
      ? LEVEL
LH 90 POKE 53248,XPOS:PD0$(Y
      POS,YPOS+LEN(SPIDER$))
      =SPIDER$
PO 98 COUNTER=0
DP 99 REM *** MAIN LOOP STAR
      T ***
CA 100 S=STICK(0)
NC 110 XPOS=XPOS+4*(S=7)-4*(
       S=11)
OC 144 IF XPOS>XMAX THEN XPO
       S=XMAX
NO 146 IF XPOS<XMIN THEN XPO
       S=XMIN
```

ATARI

Atari Inc. has cut all hardware and software prices. Please call for latest prices! ATARI is a trademark of ATARI INC

SUPERPRINTER PACKAGES

Panasonic 1091 and U-Print A	364
Panasonic 1090 and U-Print A	279
Legend 880 and U-Print A	309
Prowriter and U-Print A	385
SG-10 Printer and U-Print A	313
Citoh 7500AP and U-Print A	292
SD-10 and U-Print A	449

No additional shipping or charge card surcharges in Cont USA

MOSAIC

48K RAM	79.95
64K RAM/400	134.95
64K RAM/800 + Cable Kit #1	144.95
64K Expander for 600 XL	99.95

ATARI INTERFACES

Aid Interfast I	Call
U-Print A	64.95
R-Verter Modem Adapt	39.95
MPP 1150	Call

INDUS GT

for Atari	229

ATARI MODEM SPECIAL

Hayes 300 & R-Verter	239

MPP 1000E

Modem	109

U-PRINT BUFFERED INTERFACE

for Atari	Call

ATARI SOFTWARE

BRODERBUND

Bank Street Writer-D	49.95
Loderunner-D	23.95
Spelunker-D	20.95
Stelth-D	20.95
Print Shop-Q	34.95
Serpent's Star-D	27.95
Print Shop Paper	16.95

DATASOFT

Bruce Lee-D/T	23.95
Conan the Barbarian-D/T	27.95
Mr. Do-D/T	27.95
Alternate Reality-D	27.95
Letter/Spell Wizard-D	54.95

ELECTRONIC ARTS

Archon-D	29.95
Pinball Construction-D	29.95
M.U.L.E.-D	29.95
Murder/Zinderneuf-D	29.95
One on One-D	29.95
Archon II-D	29.95
Financial Cookbook-D	37.95
Music Construction-D	29.95
Realm/Impossibility-D	29.95
Hard Hat Mack-D	29.95

EPYX

Ballblazer-D	27.95
Return to Fractulus-D	27.95

INFOCOM

Deadline-D	29.95
Enchanter-D	23.95
Planetfall-D	24.95
Sorcerer-D	29.95
Witness-D	29.95
Cutthroats-D	24.95
Suspect-D	24.95
Hitchiker-D	24.95
Zork I-D	24.95
Zork II or III-D	27.95

MICROPROSE

F-15 Strike Eagle-D	23.95
Crusade in Europe-D	27.95
Decision in Desert-D	27.95
Kennedy Approach-D	23.95

MISCELLANEOUS ATARI

Star Leag. Baseball-D/T	23.95
Star Bowl Football-D/T	23.95
Master Type-D/Cart	27.95
Flight Simulator II-D	37.95
S.A.M.-D	41.95
Megafont-D	19.95
Monkey Wrench II-Cart	29.95
Ultima III-D	41.95
Miner 2049er-Cart	9.95
Beachhead-D/T	23.95
Spy vs. Spy-D	23.95
MPP Modem Driver-D	19.95
Net Worth-D	54.95

MISC. ATARI (cont'd.)

Omnitrend Universe-D	69.95
Adventure Writer-D	27.95
MMG Compiler-D	69.95
Summer Games-D	27.95
Pitstop II-D	27.95
Ultima I-D	23.95
Ultima II-Q	41.95
Letter Perfect/Spell-D	49.95
Ultima IV-D	41.95
Raid Over Moscow-D	27.95
Micro-Leag. Baseball-D	29.95
Paper Clip-D	59.95
Home Pak-D	37.95
Bounty Bob Strikes Back-Cart	34.95
B-Graph-D	49.95
Halley Project-D	29.95
Astral Rift-D	27.95
Back to Basic Accounting-D	Call
Data Perfect-D	49.95
QBert-Cart	24.95
Popeye-Cart	24.95
Star Wars-Cart	24.95

OPTIMIZED SYSTEMS

Action-Cart	69.95
Basic XL-Cart	54.95
MAC/65-Cart	69.95
MAC/65 Tool Kit-D	27.95
Action Tool Kit-D	27.95
Writer's Tool-Cart	69.95

SSI

Carrier Force-D	41.95
Combat Leader-Q/T	27.95
Cosmic Balance II-D	27.95
Cosmic Balance-D	27.95
Broadsides-D	27.95
War in Russia-D	55.95
50 Mission Crush-D	27.95
Questron-D	34.95
Rails West-D	27.95
Computer Ambush-D	41.95
Computer Baseball-D	27.95
Breakthrough Ardennes-D	41.95
Field of Fire-D	27.95
Imperial Galactian-D	27.95
Comp. Quarterback-D	27.95
Oper. Market Garden-D	34.95
Kampfgrupee-D	41.95

SYNAPSE

Synfile-D	34.95
Syncalc-D	34.95
Syntrend-D	34.95
Synchron-D	27.95
Syncomm-D	27.95
Synstock-D	27.95
Relax-D	79.95
Blue Max 2001-D	23.95

GENERAL HARDWARE

star MICRONICS INC
THE POWER BEHIND THE PRINTED WORD.

SG-10	
SG-15	Please Call
SD-10	For
SD-15	Current
SR-10	Prices
SR-15	
Powertype	

P.Q. the Party Quiz For Apple, C-64 & Atari	19.95

PRINTER BUFFERS

Microfazer	Call
MPP Micro Stuffer	124.95
U-Print-16K	84.94
U-Print-32K	Call
U-Print-64K Buffer	Call

MODEMS

Volksmodem 300	59
Volksmodem 1200	189
Hayes 1200	Call
Hayes 300	195
Micromodem IIE	Call
Novation	Call

PRINTERS

Citoh 8510AP	309
Legend 1080	259
Citoh 7500AP	219
Epson	Call
Toshiba 1351	Call
Toshiba 1340	649
Legend 880	229
Legend 1080	259
Panasonic 1090	209
Panasonic 1091	285
Panasonic 1092	419

APPLE

BRODERBUND

Print Shop-D	33.95
Karateka-D	21.95
Champ Loderunner-D	21.95
Bank Street Writer-D	44.95
Print Shop Graphics-D	17.95
Dazzle Draw	39.95

ELECTRONIC ARTS

Skyfox	29.95
One on One	29.95
Seven Cities/Gold	29.95
Archon II	29.95
Adv. Contruction Kit	37.95

APPLE ENTERTAINMENT

Ghostbusters	25.95
Flight Simulator II	34.95
F-15 Strike Eagle	21.95
Summer Games	25.95
Micro League Baseball	29.95
Star League Baseball	23.95
Wizardry/Proving	33.95
Ultima III	39.95
Bruce Lee	25.95
Millionaire	33.95
Death/Carnbean	21.95
Felony	21.95
Conan	25.95
Sargon III	33.95

APPLE BUSINESS

Dollars & Sense	64.95
PFS File	79.95
PFS Write	79.95
PFS Report	79.95
Multiplan	84.95
Supercalc III	124.95
Sideways	39.95
Crosstalk	129.95
Jane w/Mouse	129.95
BPI Software	Call

APPLE EDUCATION

Mastertype	25.95
Build-A-Book	25.95
Typing Tutor III	33.95
Success w/Math	19.95
Newsroom	33.95
Charlie Brown ABC	25.95
Homeword	44.95
Math Blaster	33.95
Word Attack	33.95
Winnie the Pooh	25.95
Animation Station	69.95
Koala Pad	79.95

INFOCOM

Hitchiker	25.95
Zork I, II or III	25.95
Enchanter	25.95
Deadline	33.95

SSI

Gemstone Warrior	21.95
Kampfgruppe	39.95
Phantasie	25.95
Questron	33.95

EST. 1982

ComputAbility

P.O. Box 17882, Milwaukee, WI 53217
ORDER LINES OPEN
Mon-Fri 11 a.m. - 7 p.m. CST • Sat. 12 p.m. - 5 p.m. CST

To Order Call Toll Free

800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders

414-351-2007

ORDERING INFORMATION: Please specify system. For fast delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 2 weeks to clear. Charges for COD are $3.00. School Purchase Orders welcome. In CONTINENTAL USA, include $3.00 shipping per software order. Include 3% shipping on all Hardware orders, minimum $3.00. Mastercard & Visa please include card # and expiration date. WI residents please add 5% sales tax. HI, AK, FPO, APO, Canadian orders — add 5% shipping, minimum $5.00. All other foreign orders, please add 15% shipping, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice.

COMMODORE 64

SUPER PRINTER PACKAGES

Panasonic 1091 and Cardco G+	364
Panasonic 1090 and Cardco G+	295
Legend 880 and Cardco G+	309
Prowriter and Cardco G+	389
SG-10 and Cardco G+	319

MODEMS

Westridge AA/AD	69.95
Total Tele Modem	49.95
Mighty Mo	59.95
1660 Modem	49.95

INDUS

Indus GT for C-64	239

PRINTER INTERFACES

Axiom	Call
U-Print-C	64.95
Cardco G+	69.95
Turboprint GT	69.95
MW-3	Call

C-64 MODEM SPECIAL

Westridge Modem + VIP Terminal at	99
Total Modem + VIP Terminal	84
Mighty Mo + VIP Terminal	93

Animation Station

for C-64	59.95

CALL FOR LATEST PRICES ON C-64 & C-128 HAROWARE.

COMMODORE SOFTWARE

ACCESS

Beachhead-D/T	23.95
Raid/Moscow-D/T	27.95
Scrolls of Abadon-D/T	23.95
Mach II-Cart	23.95
Beachead II-D	27.95

BATTERIES INCLUDED

Consultant-D	64.95
Paperclip w/Spellpak-D	79.95
Super Busscard II	Call
Home Organizer Items-D	16.95
B.I. 80 Card	Call
Cal-Kit	34.95
Home Pak-D	37.95

Cardco

Cardprint/B	47.95
Cassette Recorder	37.95
Write Now-Cart	34.95
File Now-D	29.95
Graph Now-D	29.95
Spell Now-D	29.95

COMMODORE

Easy Calc-D	34.95
Easy Mail-D	19.95
Easy Script-D	44.95
Easy Spell-D	19.95
Logo-D	49.95
The Manager-Q	34.95
General Ledger-D	34.95
Accts. Rec.-D	34.95
Accts. Pay.-D	34.95
Magic Desk-D	39.95
Suspend-D	29.95
Starcross-D	29.95
Deadline-D	29.95
Magic Voice	54.95
Soccer-Cart	22.95
Cutthroats-D	29.95
The Suspect-D	29.95
Inventory Man-D	34.95
Simon's Basic Cart	29.95

DATASOFT

Conan/Barbarian-D/T	27.95
Bruce Lee-D/T	27.95
Lost Tomb-D/T	23.95
Alternate Reality-D/T	27.95

ELECTRONIC ARTS

Mail Order Monster-D	29.95
Adventure Construction-D	37.95
Racing Destruction-D	29.95
Standing Stones-D	29.95
Skyfox-D	29.95

See Atari Section for rest of items and prices

EPYX

Ballblazer-D	27.95
Rescue/Fractulus-D	27.95
F.B.I.-D	27.95
Fast Load Cart	27.95

INFOCOM

Call for items and prices

T-Cassette
D-Disk
Cart-Cartridge

MISC. COMMDDDRE

Ultima III-D	41.95
Flight Simulator II-D	37.95
Star Leag. Baseball-D/T	23.95
Mastertype-D/Cart	27.95
Strip Poker-D	23.95
Sam-D	41.95
Spelunker-D	20.95
Stealth-D	20.95
Bungling Bay-D	20.95
Dr. Creep-D	20.95
On-Field Tennis-D/T	23.95
F-15 Strike Eagle-D/T	23.95
Movie Maker-D	37.95
VIP Terminal-D	29.95
Doodle-D	27.95
Ghost Busters-D	24.95
Koala Printer-D	19.95
Karateka-D	20.95
Print Shop-D	34.95
Amazon-D	22.95
Rendezvous/Rama-D	22.95
Farenheit 451-D	22.95
Dragonworld-D	22.95
Shadow Keep-D	22.95
Spy vs. Spy-D	23.95
Stress Relax Sys.-D	79.95
Blue Max-D/T	23.95
Ultima IV-D	41.95
Ultima II-D	41.95
Micro-League Baseball-D	29.95
Home Pak-D	37.95
Math Blaster-D	34.95
Word Attack-D	34.95
Speed Reader II	49.95
Stunt Flyer-D	Call
Sargon III-D	34.95
Super Zaxxon-D	23.95
Decision/Desert-D	27.95
Kennedy Approach-D	23.95
Crusade/Europe-D	27.95
Halley Project-D	29.95
Transylvannia-D	23.95
Bounty Bob-Cart	34.95
Blue Max 2001-D	23.95
Fleet System II-D	49.95
Evelyn Wood-D	34.95

SSI

Carrier Force-D	41.95
Questron-D	27.95
Broadsides-D	27.95
Comp. Quarterback-D	27.95
Computer Ambush-D	41.95
Kamptgruppe-D	41.95
Oper. Market Garden-D	34.95
Gemstone Warnor-D	23.95
Imperium Galactium-D	27.95
Phantasie-D	27.95
Cartels/Cutthroats-D	27.95

SPINNAKER

Call for items and prices

SYNAPSE

Call for items and prices

TIMEWORKS

Call for items and prices

No surcharge for MasterCard or Visa

SPECIAL ISSUE

COMPUTE!'s Apple Applications

COMPUTE!'s Apple Applications issue features applications, tutorials, and in-depth feature articles for owners and users of Apple personal computers. From software to hardware to the state of the industry, this special issue serves as a useful tool and handy reference. It's filled with home, business, and educational applications and purchasing information.

Like *COMPUTE!* and *COMPUTE!'s Gazette*, *COMPUTE!'s Apple Applications* issue contains ready-to-type-in programs, easy-to-understand tutorials, and useful information. This special issue includes something for everyone:

ON SALE: April 9, 1985
PRICE: $3.95 (In Canada, $4.75)

FEATURES

Business Applications Software: A Buyer's Guide
Word processing, spreadsheets, databases, and more

Telecommunicating on the Apple II and the Macintosh
"Is Anyone Out There? Getting Started with Telecommunications," a handy reader introduction to the first few hours of telecommunications, plus "Exploring Databases" by computer

A Game-Lover's Choice
"The Ten Best-Ever Apple Games"

Apple in the Coming Years
"Up and Coming Apple," watchers and pragnasticators reveal their best crystal-ball impressions of the market, the manufacturer, and the future development of our industry

Apple in Education
"The State of Educational Computing," plus "A Software Overview: What's Available in Education?" and "What Makes Superior Educational Software?"

APPLICATIONS

Database Management Program
A small, working program for the Apple II series

Chess
A significant user-ready program with five levels of challenging play

Softsearcher
A machine language search routine

Inside a MacArtist
A major applications feature on the inner workings of *MacPaint*, *MacDraw*, and *MacWrite* with tutorials

Stargazer
Guess the constellation

Spelling Bee
Word-guessing game for youngsters

Heat Seeker
A fast-paced arcade style game

The programs published in *COMPUTE!'s Apple Applications* are available on disk, ready to run on your Apple IIc, Apple II+, and IIe. The *Disk* will save you hours of typing time and give you easy access to the programs in this special issue of *COMPUTE!'s Apple*. The *Disk* is only $12.95 and is available only through COMPUTE! Publications. So order your *Disk* today!

Look for the Spring/Summer issue of *COMPUTE!'s Apple Applications* on sale where you buy *COMPUTE!* and *COMPUTE!'s Gazette*, and at Apple computer retailers. Or order directly from COMPUTE!.

Send in the attached card with your payment or call toll-free **1-800-334-0868**.

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies
324 W. Wendover Avenue, Suite 200 Greensboro, NC 27408 919-275-9809

Publishers of *COMPUTE!*, *COMPUTE!'s Gazette*, *COMPUTE!'s Gazette Disk*, COMPUTE! Books, and *COMPUTE!'s Apple Applications*

```
EB 150  POKE 53248,XPDS
CM 200  S=STRIG(0)
OC 210  IF S<>0 THEN 400
FH 211  REM ** WEBSTER LEAPS
CJ 212  PDKE 5327B,0
OC 214  VEL=XB3VEL:XB3VEL=VEL
        /2
FL 220  FDR I=YPDS TO YB-4 ST
        EP 4
HP 240  PO0$(I,I+LS)=SPIDER$
LF 244  SOUNO 0,I,10,4
MN 246  IF X83=0 THEN FOR D=1
         TD 6:NEXT D:GOTD 250
LJ 248  GOSU8 B30
BP 250  NEXT I
PI 260  HIT=PEEK(53260):IF HI
        T<>0 THEN GDSUB 1000
IO 280  FDR I=YB-4 TD YPDS ST
        EP -4
HM 300  PO0$(I,I+LS)=SPIOER$
LC 304  SDUND 0,I,10,4
MH 306  IF XB3=0 THEN FDR D=1
         TO 6:NEXT D:GDTD 310
LG 308  GDSUB 830
BM 310  NEXT I
GC 320  SOUND 0,0,0,0
GP 322  XB3VEL=VEL
FH 399  REM ** BUG 1 MDVEMENT
FD 400  IF XB1<>0 THEN 430
OP 410  X=INT(51*RND(0)-25)
BG 414  IF ABS(X)<>1 THEN 500
PJ 418  X81=XMIN*(X=1)+XMAX*(
        X=-1)
EL 419  XB1VEL=X*XB1SPEEO
NC 424  PO1$(Y8,YB+LEN(8UG$))
        =8UG$
HB 430  SOUNO 1,30,14,4
EI 440  X81=XB1+X81VEL
LL 442  IF X81<XMIN OR XB1>XM
        AX THEN X81=0
MG 450  POKE 53249,XB1
GL 490  SOUNO 1,0,0,0
FJ 499  REM ** 8UG 2 MOVEMENT
GB 500  IF XB2<>0 THEN 530
PA 510  X=INT(51*RND(0)-25)
BI 514  IF A8S(X)<>1 THEN 600
PL 518  X82=XMIN*(X=1)+XMAX*(
        X=-1)
ED 519  X82VEL=X*XB2SPEEO
HE 524  PO2$(Y8,Y8+LEN(BUG$))
        =8UG$
NC 530  SOUNO 2,11,14,4
EM 540  X82=X82+X82VEL
LP 542  IF X82<XMIN OR XB2>XM
        AX THEN X82=0
MA 550  POKE 53250,X82
GN 590  SOUND 2,0,0,0
FL 599  REM ** 8UG 3 MOVEMENT
GE 600  IF XB3<>0 THEN 630
BD 610  X=INT(F1*RNO(0)-F2)
BE 614  IF ABS(X)<>1 THEN 100
PJ 616  COUNTER=COUNTER+1:IF
        COUNTER>15 THEN POP :
        GOTO 50
PN 618  X83=XMIN*(X=1)+XMAX*(
        X=-1)
FB 619  X83VEL=X*XB3SPEED
NG 624  PD3$(Y8,Y8+LEN(BUG$))
        =8UG$
LE 630  GOSU8 B30
GB 700  GOTO 100
EE 702  REM *** ENO OF MAIN L
        OOP ***
CD 829  REM ** MOVE 8UG 3
HM 830  SOUNO 3,5,8,4
FC 840  X83=X83+X83VEL
MF 842  IF X83<XMIN OR X83>XM
        AX THEN XB3=0
MF 850  POKE 53251,X83
DF 860  IF X83=X81 THEN X81=0
        :POKE 53249,0
DB 862  IF XB3=X82 THEN XB2=0
        :PDKE 53250,0
HB B90  SDUND 3,0,0,0
IK B99  RETURN
HD 1000 REM SPIOER HIT A BUG
        (WHICH?)
GJ 1004 HIT=INT(HIT/2)
CP 1010 DN HIT GDSUB 1100,12
        00,1300,1400,1500,16
        00,1700
KD 1011 RETURN
LG 1100 REM SPIDER ATE BUG 1
GC 1110 XB1=0:PDKE 53249,XB1
IL 1120 SCDRE=SCDRE+25
BC 1125 GDSUB 1B00
KF 1130 RETURN
LI 1200 REM SPIDER ATE BUG 2
FN 1210 XB2=0:POKE 53250,XB2
IN 1230 SCORE=SCORE+25
BE 1235 GDSUB 1B00
KH 1240 RETURN
NA 1300 REM SPIDER ATE BUGS
        1 AND 2
AM 1310 GOSUB 1100
AJ 1320 GDSUB 1200
IN 1330 SCDRE=SCDRE+50
BF 1335 GDSUB 1B00
KI 1340 RETURN
NN 1400 REM HIT BAD BUG (5)
        - TOO BAD
DK 1410 FOR I=20 TO 200 STEP
        4
DK 1420 SOUNO 0,I,14,8
EA 1430 SETCOLOR 4,I,8
DH 1440 FOR N=1 TO 10:NEXT N
FC 1450 NEXT I
GB 1455 POKE 656,1:POKE 657,
        LIVES*2+16:? " "
CJ 1460 SOUNO 0,0,0,0:SETCOL
        OR 4,0,0
GM 1470 LIVES=LIVES-1
KD 1480 IF LIVES<1 THEN GOTO
         19000
KD 1490 RETURN
JC 1500 REM SPIDER ATE BUG 3
        AND BAD BUG
GG 1510 X81=0:POKE 53249,X81
AN 1520 GOSUB 1400
KJ 1530 RETURN
JE 1600 REM SPIDER ATE BUG 2
        AND BAD BUG
GB 1610 X82=0:POKE 53250,XB2
AD 1620 GOSU8 1400
KK 1630 RETURN
DH 1700 REM SPIDER ATE ALL 3
        UGS
GI 1710 X81=0:POKE 53249,XB1
GD 1720 X82=0:POKE 53250,X82
BA 1730 GOSU8 1400
KM 1740 RETURN
AM 1800 REM UPDATE SCORE
JE 1810 POKE 656,2:POKE 657,
        32
NE 1820 PRINT SCORE
KN 1830 SOUNO 0,40,12,8:FOR
        I=1 TO 10:NEXT I:SOU
        NO 0,0,0,0
LL 1899 RETURN
MD 7000 REM INITIALIZATION
KA 7010 OIM X$(1)
DL 7020 A=AOR(X$)
BP 7030 8=INT((A-512)/1024+1
        )*1024
NC 7040 OIM F$(8-A+511)
IA 7050 OIM PO0$(128),PO1$(1
        28),PO2$(128),PO3$(1
        2B)
HN 7060 POKE 54279,INT(B/256
        )
LJ 7099 RETURN
MB 7100 REM CONSTRUCT CHOOSE
        HERE
EF 7110 OIM SPIOER$(15)
MB 7120 FOR I=1 TO 15
DF 7130 READ X:SPIOER$(I)=CH
        R$(X)
FE 7140 NEXT I
GE 7150 DATA B,8,B,B,137,74,
        60,255,60,74,129,0,0
        ,0,0
MG 7152 LS=LEN(SPIDER$)
CD 7160 OIM 8UG$(3)
ED 7170 FOR I=1 TD 3
AB 7180 READ X:BUG$(I)=CHR$(
        X)
FJ 7190 NEXT I
HD 7200 OATA 2B,62,42
KE 7300 REM ** INTRO OISPLAY
LB 7310 PDKE 5324B,0:POKE 53
        249,0:POKE 53250,0:P
        DKE 53251,0
DB 7320 GRAPHICS 2+16
CE 7330 ? #6:? #6:? #6;"
        {6 SPACES}WEBSTERS"
PE 7340 ? #6:? #6:? #6;"
        {5 SPACES}DINES  DUT
        "
LH 7350 ? #6:? #6:? #6:? #6;
        "{4 SPACES}press  st
        art"
PK 7379 REM ** PLAY MUSIC-AW
        AIT START
PB 7380 RESTORE 19100
HD 7384 POKE 53279,8
JG 7390 FOR I=1 TO 73:READ X
        :SOUNO 2,X,10,8
JI 7392 IF PEEK(53279)=6 THE
        N POP :SOUNO 2,0,0,0
        :RETURN
NF 7394 FOR J=1 TO 7:NEXT J:
        NEXT I
EB 7395 FOR J=1 TO 40:NEXT J
DE 7396 GOTO 7380
LC 7600 REM GAME DISPLAY SET
        UP
FE 7610 GRAPHICS 5
CL 7620 REM CLEAR COLLISION
        REGISTER
GE 7630 POKE 53278,0
NH 7640 REM SET PLAYER SIZES
        (2X)
MH 7650 POKE 53256,0:PDKE 53
        257,0:POKE 53258,0:P
        OKE 53259,1
EC 7660 REM SET PLAYER COLOR
        S
JG 7670 POKE 704,10:POKE 705
        ,255:POKE 706,148:PO
        KE 707,255
HN 7680 REM ENABLE P/M GRAPH
        ICS
GN 7690 POKE 53277,3
HG 7700 REM SET 2-LINE GRAPH
        ICS
DH 7710 POKE 559,46
LG 7720 REM TURN OFF CURSOR
PN 7722 POKE 752,1
FP 7740 REM ZERO OUT CHARS
BB 7750 PO0$=CHR$(0):PD0$(12
        8)=CHR$(0):PD0$(2)=P
        D0$
HF 7760 PO1$=PD0$:PO2$=PD0$:
        PO3$=PO0$
DK 7809 REM ** INITIAL SPIDE
        R LOCATION
AN 7810 XPOS=100:YPOS=21
BC 7811 YB=90
GD 7812 REM ** INIT BUG POS
        & VELS
PD 7814 XB1=0:XB2=0:XB3=0
KK 7816 XB1VEL=0:XB2VEL=0:XB
        3VEL=0
FB 7817 X81SPEEO=1:XB2SPEED=
        1:X83SPEED=6
```

```
PD 7818  XMIN=42:XMAX=200
EB 7820  F1=51:F2=25
GC 7900  REM DRAW BACKGROUND
LJ 7912  SETCOLOR 0,12,6:SETC
         OLOR 1,14,2:SETCOLOR
          2,0,6
GM 7913  REM ** DRAW LAWN
IF 7914  COLOR 1
DO 7920  PLOT 0,38:DRAWTO 79,
         38:PLOT 0,39:DRAWTO
         79,39
JD 7928  COLOR 2:PLOT 44,7:DR
         AWTO 60,2
BP 7930  COLOR 2:POKE 765,2:R
         ESTORE 7940
ND 7932  FOR I=1 TO 4:READ X,
         Y:PLOT X,Y
GK 7933  FOR N=1 TO 3:READ X,
         Y:DRAWTO X,Y
LC 7934  NEXT N:READ X,Y:POSI
         TION X,Y:XIO 18,#6,0
         ,0,"S:":NEXT I
JA 7940  DATA 74,30,79,30,79,
         37,71,37,74,30,74,7,
         79,7,79,30,74,30,74,
         7,70,0,73,0,79,7,74,
         7,70,0,77,0,79,0
LE 7942  DATA 79,7,77,5,77,0
EG 7946  PLOT 0,0:DRAWTO 0,2:
         POSITION 0,0:XIO 18,
         #6,0,0,"S:"
EE 7949  REM ** PLOT TREE LEA
         VES
IF 7950  COLOR 1
AA 7952  RESTORE 7954:FOR I=1
          TO 24:READ X,Y:PLOT
          X,Y:NEXT I
LO 7954  DATA 1,3,1,4,2,3,3,4
         ,3,5,7,2,2,2,7,3,8,4
         ,3,2,6,2,10,2,3,0,4,
         0,4,1,4,2,5,0,6,0,6,
         2,7,2,45,8,44,5,45,6
         ,58,4
AC 7980  POKE 623,4
AO 7990  LIVES=3:LEVEL=0:SCOR
         E=0
FO 7992  POKE 656,1:POKE 657,
         10:? "LIVES
         {3 SPACES}* * *"
JL 7996  POKE 656,2:POKE 657,
         2:? "LEVEL:
         {17 SPACES}SCORE:"
CJ 7997  GOSUB 1800

MC 7999  RETURN
JB 10000 REM LEVEL 2 ADJUSTM
         ENTS
JO 10010 XB1SPEED=3
KD 10019 REM ** DRAW A BIG R
         OCK
KF 10020 COLOR 3
JO 10030 PLOT 11,37:DRAWTO 2
         5,37:PLOT 12,36:ORA
         WTO 25,36:PLOT 13,3
         5:DRAWTO 20,35:PLOT
          22,35:DRAWTO 24,35
OO 10040 PLOT 14,34:DRAWTO 1
         8,34
GD 10050 F1=31:F2=15
OO 10099 RETURN
OH 10100 REM **LEVEL 3 ADJUS
         TMENTS
NM 10104 REM ** ORAW GRASS
PF 10117 N=20:HEIGHT=1:GOSUB
          12000
EL 10119 REM ** DRAW A SMALL
          ROCK
KH 10120 COLOR 3:PLOT 60,37:
         ORAWTO 65,37:PLOT 6
         2,36:DRAWTO 64,36
KB 10130 XB2SPEED=2
OE 10199 RETURN
OJ 10200 REM **LEVEL 4 ADJUS
         TMENTS
PD 10205 N=10:HEIGHT=2:GOSUB
          12000
PA 10210 N=10:HEIGHT=3:GOSUB
          12000
DF 10299 RETURN
OL 10300 REM **LEVEL 5 ADJUS
         TMENTS
FM 10310 F1=21:F2=10
PC 10320 N=10:HEIGHT=3:GOSUB
          12000
DG 10399 RETURN
DN 10400 REM **LEVEL 6 ADJUS
         TMENTS
HI 10410 XB1SPEED=4:XB2SPEED
         =4:XB3SPEED=8
KM 10420 N=10:HEIGHT=3:GOTO
          12000
OL 10430 F1=17:F2=8
DH 10499 RETURN
HN 12000 REM ** TO ORAW SOME
          TALL GRASS
KE 12010 COLOR 1
IE 12020 FOR I=1 TO N
LO 12030 X=INT(80*RND(0))
AD 12040 PLOT X,37:ORAWTO X,
         37-HEIGHT
IA 12050 NEXT I
NJ 12060 RETURN
AL 19000 REM * END GAME & L
         OSS SCORE**
DF 19005 POKE 53248,0:POKE 5
         3249,0:POKE 53250,0
         :POKE 53251,0
BJ 19010 GRAPHICS 2+16:? #6;
         "{5 SPACES}game ove
         r ":? #6:? #6;"
         {4 SPACES}SCORE : "
         ;SCORE:? #6
EH 19020 ? #6:? #6;" D=SIG
         N DOES YOU"
FP 19030 ? #6:? #6:? #6:? #6
         ;"{6 SPACES}SUCCESS
         P.S"
BN 19040 RESTORE 19100
FI 19050 FOR I=1 TO 73:READ
         X
GG 19060 SOUND 2,X,10,8:FOR
         J=1 TO 5:NEXT J:NEX
         T I
IL 19099 REM MUSIC DATA
MK 19100 DATA 162,0,121,0,0,
         96,0,121,0,0,162,0,
         121,0,0,96,0,121,0,
         0,162,0,121,0,0,108
         ,96,0,108,0,121,0,1
         28,0,0,121
DF 19110 DATA 108,0,0,162,0,
         128,0,0,108,0,128,0
         ,0,162,0,128,0,0,10
         8,0,128,0,0,162,0,8
         1,72,81,91,96,108,1
         21,0,0
MM 19120 DATA 108,121,0
JP 19200 GRAPHICS 2+16:SETCO
         LOR 4,4,2:SETCOLOR
         0,12,6:SETCOLOR 1,8
         ,4:SETCOLOR 2,0,8
CM 19210 POSITION 4,4:? #6;"
         PlAy AgAiN (Y/N)"
AD 19220 ? #6:? #6;"
         {7 SPACES}(Y/N)"
JH 19230 POKE 764,255
LN 19240 I=PEEK(764):IF I<>3
         5 AND I<>43 THEN 19
         240
NI 19250 IF I=35 THEN GOTO 2
         0000
HA 19260 POKE 764,255:GOTO 3
         0
MJ 20000 END
```

Program 2: Webster Dines Out For Commodore 64

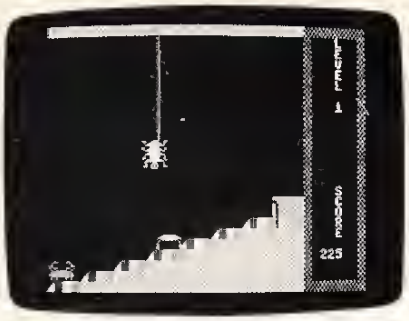
Translation by Jeff Hamdani

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
100 GOSUB1130:GOSUB1150:GOSUB1
    190:GOSUB1580:GOSUB1370
                          :rem 233
110 LV=LV+1:CR=0:IFLV=7THEN149
    0                     :rem 135
120 ONLVGOSUB130,1400,1450,146
    0,1470,1480:IFCR>10THEN110
                          :rem 115
130 POKE214,6:PRINT:POKE211,35
    :PRINTLV              :rem 125
140 POKE214,20:PRINT:POKE211,3
    3:PRINTSC             :rem 156
160 P=PEEK(56320):D=15-(PAND15
    )                     :rem 194
170 XP=XP+8*(D=4)-8*(D=8)
                          :rem 81
180 YP=INT((-.208)*XP+227.2)
                          :rem 249
190 IFXP>226THENXP=226 :rem 87
200 IFXP<25THENXP=25   :rem 231
210 POKEV,XP:POKEV+1,F:rem 121
220 FR=PAND16          :rem 243
230 IFFR<>0THEN400     :rem 41
240 VE=V3:V3=VE/2:POKET1,17
                          :rem 117
250 XX=INT(XP/8)-1:FORI=PTOYPS
    TEP8:POKEV,XP:POKEV+1,I
                          :rem 253
260 O=1064+XX+Z:POKEH1,I:POKEL
    1,(I/3)               :rem 180
270 POKEO,66:POKEO+CL,6
                          :rem 131
280 IFB3=0THEN300      :rem 205
290 GOSUB750           :rem 183
300 Z=Z+40:NEXTI       :rem 213
310 A=PEEK(V+30)AND15:IF(AAND1
    )=1THENGOSUB860:IFLS=0THEN
    POKET1,16:GOTO1030 :rem 65
320 FORI=YPTOPSTEP-8:POKEO,32
                          :rem 126
330 POKEV,XP:POKEV+1,I:POKEH1,
    (I/3):POKEL1,I        :rem 228
340 IFB3=0THEN360      :rem 208
350 GOSUB 750          :rem 180
360 O=O-40:NEXTI:Z=0:POKET1,16
                          :rem 73
370 REM MUSI           :rem 188
380 V3=VE              :rem 252
390 REM BUG1 MOVEMENT :rem 250
400 IFB1<>0THEN460       :rem 9
410 POKEV+21,PEEK(V+21)OR2
                          :rem 1
420 X=INT(F1*RND(0)-F2):rem 18
430 IFABS(X)<>1THEN520 :rem 22
440 B1=(-1)*(MI*(X=1)+MA*(X=-1
    ))                    :rem 112
450 V1=X*S1             :rem 99
460 B1=B1+V1:POKET1,17:rem 241
470 IFB1<MIORB1>MATHENB1=0:POK
    ET1,16                :rem 223
480 Y1=INT((-.208)*B1+233.2)
                          :rem 165
490 POKEV+2,B1:POKEV+3,Y1:POKE
    V+40,2                :rem 157
500 POKEH1,INT(RND(0)*3+19):PO
    KEL1,INT(RND(0)*1+9):POKET
    1,16                  :rem 239
510 REM BUG2 MOVEMENT :rem 245
520 IFB2<>0THEN580       :rem 16
530 POKEV+21,PEEK(V+21)OR4
                          :rem 6
```

```
540 X=INT(F1*RND(0)-F2):rem 21
550 IF ABS(X)<>1THEN640:rem 28
560 82=(-1)*(MI*(X=1)+MA*(X=-1
    ))                  :rem 116
570 V2=X*S2             :rem 104
580 B2=82+V2:POKET2,17:rem 248
590 IFB2<MIOR82>MATHENB2=0:POK
    ET2,16              :rem 230
600 Y2=INT((-.208)*82+233.2)
                        :rem 161
610 POKEV+4,B2:POKEV+5,Y2:POKE
    V+41,3              :rem 159
620 POKEH2,INT(RND(0)*3+25):PO
    KEL2,INT(RND(0)*1+9):POKET
    2,16                :rem 242
630 REM BUG3 MOVEMENT   :rem 249
640 IFB3<>0THEN710      :rem 15
650 X=INT(F1*RND(0)-F2):rem 23
660 IFABS(X)<>1THEN 160:rem 27
670 CR=CR+1:IFCR>10ANDLV=1THEN
    850                 :rem 12
680 IFCR>10ANDLV>1THENGOSUB850
    :GOTO110            :rem 150
690 B3=(-1)*(MI*(X=1)+MA*(X=-1
    ))                  :rem 121
700 V3=X*S3             :rem 101
710 GOSUB750            :rem 180
720 GOTO160             :rem 105
730 REM END OF MAIN LOOP
                        :rem 73
740 REM MOVE BUG3       :rem 199
750 POKET3,129          :rem 26
760 B3=B3+V3            :rem 120
770 IFB3<MIORB3>MATHENB3=0
                        :rem 103
780 Y3=INT((-.208)*B3+233.2)
                        :rem 172
790 POKEV+6,B3:POKEV+7,Y3:POKE
    V+42,6              :rem 178
800 POKEH3,17:POKEL3,5:POKET3,
    128                 :rem 214
810 IFB3=B1ANDFR<>0THENB1=0:PO
    KEV+40,0:POKEV+2,B1:POKET1
    ,16                 :rem 87
820 IFB3=B1ANDFR=0THENB1=0:POK
    EV+40,0:POKEV+2,B1:rem 154
830 IFB3=B2THENB2=0:POKEV+41,0
    :POKEV+4,B2:POKET2,16
                        :rem 75
840 REM MUSIC            :rem 1
850 RETURN              :rem 125
860 BN=0:FORX=1TO3:BG=2↑X
                        :rem 64
870 IF(AANDBG)THENGOSUB900
                        :rem 100
880 NEXT:IFBN=2THENSC=SC+50
                        :rem 207
885 IFBN=3THENSC=SC+100
                        :rem 136
890 POKE214,20:PRINT:POKE211,3
    3:PRINTSC:RETURN    :rem 194
900 IFBG=8THENGOSUB960:RETURN
                        :rem 142
910 IFPEEK(V+2*X)>=XP-24ANDPEE
    K(V+2*X)<=XP+24THENGOSUB95
    0:SC=SC+25:POKEV+2*X,0
                        :rem 54
920 IFPEEK(V+2)=0THENB1=0
                        :rem 207
930 IFPEEK(V+4)=0THENB2=0:POKE
    T2,16               :rem 85
940 RETURN              :rem 125
950 BN=BN+1:POKEV+21,PEEK(V+21
    )AND15-BG:RETURN    :rem 51
960 IFPEEK(V+2*X)>=XP-24ANDPEE
    K(V+2*X)<=XP+24THENLS=LS-1
    :GOTO 980           :rem 219
970 RETURN              :rem 128
980 REM SPIDER HITS BIG BUG
                        :rem 52
990 FORI=8TO2STEP-2:POKE53281,
    I:POKE53280,I       :rem 120
1000 PRINT"{HOME}{WHT}
    {9 RIGHT}LIVES REMAINING
    {SPACE}";LS:NEXT    :rem 31
1010 FORI=1TO700:NEXT:FORI=102
    4TO1055:POKEI+CL,5:POKEI,
    160:NEXT            :rem 170
1020 POKE53280,0:POKE53281,0:R
    ETURN               :rem 50
1030 REM SPIDER LOST ALL LIVES
                        :rem 13
1040 O=0:XX=0:Z=0:FORI=0TO3:PO
    KEV+21,PEEK(V+21)AND(15-2
    ↑I):POKEV+2*I,0:NEXT
                        :rem 77
1050 PRINT"{CLR}{6 DOWN}"SPC(1
    5)"GAME OVER"       :rem 199
1060 8=13-LEN(STR$(SC))/2:PRIN
    T"{2 DOWN}"SPC(B)"YOUR SC
    ORE IS ";SC         :rem 142
1070 PRINT"{2 DOWN}"SPC(12)"
    {RVS}TRIGGER{OFF} TO STAR
    T"                  :rem 130
1080 PRINT"{2 DOWN}"SPC(15)"
    {RVS}N{OFF} TO STOP"
                        :rem 120
1090 FR=PEEK(56320)AND16:GETA$
    :IFA$=""ANDFR<>0THEN1090
                        :rem 139
1100 IFFR=0THEN100      :rem 22
1110 IFA$="N"THENEND    :rem 140
1120 GOTO 1090          :rem 199
1130 PRINT"{CLR}":XP=100:CL=54
    272:V=53248:SC=0:LV=0:LS=
    3:B1=0:B2=0:B3=0:F=56
                        :rem 254
1140 V1=0:V2=0:V3=0:S1=4:S2=4:
    S3=12:MI=0:MA=255:F1=51:F
    2=25:RETURN         :rem 122
1150 SD=CL:FORL=SDTOSD+24:POKE
    L,0:NEXT:T1=SD+4:D1=SD+5:
    R1=SD+6:H1=SD+1:L1=SD
                        :rem 245
1160 T2=T1+7:T3=T2+7:D2=D1+7:D
    3=D2+7:R2=R1+7:R3=R2+7:H2
    =H1+7:H3=H2+7:L2=L1+7
                        :rem 18
1170 L3=L2+7:POKE54296,15:POKE
    D1,17:POKER1,241:POKED2,1
    7:POKER2,241:POKED3,17
                        :rem 34
1180 POKER3,241:RETURN  :rem 91
1190 POKE53281,0:POKE53280,0:F
    ORI=1024TO1056:POKEI+CL,5
    :POKEI,160:NEXT     :rem 184
1200 FORI=1984TO1743STEP-36:PO
    KEI+CL,1:POKEI,233:NEXT
                        :rem 2
1210 FORI=1985TO1733STEP-36:PO
    KEI+CL,2:POKEI,160:NEXT
                        :rem 3
1220 FORI=1986TO1744STEP-36:PO
    KEI+CL,3:POKEI,160:NEXT
                        :rem 8
1230 FORI=1987TO1705STEP-36:PO
    KEI+CL,4:POKEI,160:NEXT
                        :rem 8
1240 POKE1692+CL,7:POKE1692,20
    1:FORI=1693TO1696:POKEI+C
    L,7:POKEI,160:NEXT
                        :rem 143
1250 POKE1732+CL,7:POKE1732,22
    1:FORI=1733TO1736:POKEI+C
    L,7:POKEI,160:NEXT
                        :rem 126
1260 POKE1772+CL,7:POKE1772,22
    1:FORI=1773TO1776:POKEI+C
    L,7:POKEI,160:NEXT
                        :rem 143
1270 FORI=1988TO2016:POKEI+CL,
    15:POKEI,160:NEXT:FORI=19
    52TO1976:POKEI+CL,15:
                        :rem 147
1280 POKEI,160:NEXT:FORI=1916T
    O1936:POKEI+CL,15:POKEI,1
    60:NEXT:FORI=1880TO1896:
                        :rem 134
1290 POKEI+CL,15:POKEI,160:NEX
    T:FORI=1844TO1856:POKEI+C
    L,15:POKEI,160:NEXT
                        :rem 89
1300 FORI=1808TO1816:POKEI+CL,
    15:POKEI,160:NEXT   :rem 97
1310 FORI=1056TO1063:POKEI+CL,
    12:POKEI,102:POKEI+960+CL
    ,12:POKEI+960,102:NEXT
                        :rem 80
1320 FORI=1056TO2016STEP40:POK
    EI+CL,12:POKEI,102:POKEI+
    7+CL,12:POKEI+7,102:NEXT
                        :rem 32
1330 RESTORE:FORI=1100TO1260ST
    EP40:READA:POKEI+CL,1:POK
    EI,A:NEXT           :rem 88
1340 FORI=1660TO1820STEP40:REA
    DA:POKEI+CL,1:POKEI,A:NEX
    T                   :rem 8
1350 DATA12,5,22,5,12,19,3,15,
    18,5                :rem 164
1360 RETURN             :rem 170
1370 POKEV+21,9:POKE53275,14
                        :rem 67
1380 FORI=0TO3:POKEI+2040,192+
    I:NEXT              :rem 151
1390 RETURN             :rem 173
1400 REM LEVEL 2 ADJUSTMENTS
                        :rem 165
1410 S1=5:F1=31:F2=15   :rem 99
1420 FORI=1904TO1907:POKEI+CL,
    INT(RND(0)*15+1):POKEI,16
    0:NEXT              :rem 137
1430 FORI=1944TO1947:POKEI+CL,
    INT(RND(0)*15+1):POKEI,16
    0:NEXT              :rem 146
1440 FORI=1984TO1987:POKEI+CL,
    INT(RND(0)*15+1):POKEI,16
    0:NEXT:RETURN       :rem 181
1450 S1=6:F1=31:F2=15:RETURN
                        :rem 130
1460 S2=6:RETURN        :rem 221
1470 S1=8:S3=16:F1=21:F2=10:RE
    TURN                :rem 228
1480 S1=10:S2=10:S3=20:F1=17:F
    2=8:RETURN          :rem 66
1490 FORI=0TO3:POKEV+21,PEEK(V
    +21)AND(15-2↑I):POKEV+2*I
    ,0:NEXT             :rem 8
1500 PRINT"{CLR}{6 DOWN}
    {9 RIGHT}YOU WON THE GAME
    "                   :rem 129
1510 PRINT"{2 DOWN}{9 RIGHT}YO
    UR SCORE IS ";SC    :rem 247
1520 PRINT"{2 DOWN}{9 RIGHT}
    {RVS}TRIGGER{OFF} TO STAR
    T"                  :rem 169
1530 PRINT"{2 DOWN}{9 RIGHT}
    {RVS}N{OFF} TO STOP"
                        :rem 156
1540 FR=PEEK(56320)AND16:GETA$
    :IFA$=""ANDFR<>0THEN1540
                        :rem 139
1550 IFFR=0THEN100      :rem 31
1560 IFA$="N"THENEND    :rem 149
1570 GOTO 1540          :rem 208
1580 I=12288            :rem 89
1590 READ A:IF A=256 THEN RETU
    RN                  :rem 31
1600 POKE I,A:I=I+1:GOTO 1590
                        :rem 83
```

```
1610 DATA 224,126,7,96,255,6
                        :rem 103
1620 DATA 39,255,228,31,255,24
8                       :rem 1
1630 DATA 1,255,128,193,255,13
1                       :rem 246
1640 DATA 199,255,227,63,255,2
52                      :rem 57
1650 DATA 1,255,128,7,255,224
                        :rem 149
1660 DATA 31,255,252,33,255,13
0                       :rem 242
1670 DATA 97,255,134,225,255,1
34                      :rem 53
1680 DATA 15,255,243,31,255,24
8                       :rem 254
1690 DATA 16,60,4,24,126,4
                        :rem 250
1700 DATA 24,219,6,120,219,7
                        :rem 92
1710 DATA 0,126,0,0,0,0:rem 72
1720 DATA 0,0,0,0,3,255:rem 79
1730 DATA 192,31,255,248,127,2
55                      :rem 49
1740 DATA 254,151,255,233,32,0
                        :rem 189
1750 DATA 4,64,0,2,128,0
                        :rem 142
1760 DATA 1,192,0,3,192,0
                        :rem 192
1770 DATA 3,192,0,3,192,0
                        :rem 195
1780 DATA 3,192,0,3,192,0
                        :rem 196
1790 DATA 3,128,0,1,64,0
                        :rem 144
1800 DATA 2,32,0,4,16,0:rem 81
1810 DATA 8,24,0,24,252,0
                        :rem 189
1820 DATA 63,0,0,0,0,0 :rem 26
1830 DATA 0,0,3,255,192,31
                        :rem 241
1840 DATA 255,248,127,255,254,
151                     :rem 101
1850 DATA 255,233,32,0,4,64
                         :rem 43
1860 DATA 0,2,128,0,1,192
                        :rem 191
1870 DATA 0,3,192,0,3,192
                        :rem 196
1880 DATA 0,3,192,0,3,192
                        :rem 197
1890 DATA 0,3,192,0,3,128
                        :rem 197
1900 DATA 0,1,64,0,2,32:rem 82
1910 DATA 0,4,16,0,8,24:rem 90
1920 DATA 0,24,252,0,63,0
                        :rem 186
1930 DATA 15,0,240,24,129,24
                        :rem 87
1940 DATA 48,0,12,240,0,15
                        :rem 239
1950 DATA 152,129,25,207,0,243
                        :rem 194
1960 DATA 96,0,6,31,255,248
                        :rem 57
1970 DATA 63,255,254,127,255,2
54                      :rem 56
1980 DATA 255,255,255,127,255,
254                     :rem 109
1990 DATA 49,129,140,99,0,198
                        :rem 171
2000 DATA 66,0,66,132,0,33
                        :rem 236
2010 DATA 132,0,33,132,0,33
                        :rem 17
2020 DATA 66,0,66,99,0,198
                         :rem 6
2030 DATA 247,129,239,0,0,256
                        :rem 143
```



Commodore 64 "Webster Dines Out" takes advantage of sprite graphics.

"Webster Dines Out" for the VIC-20, a fast machine language game.

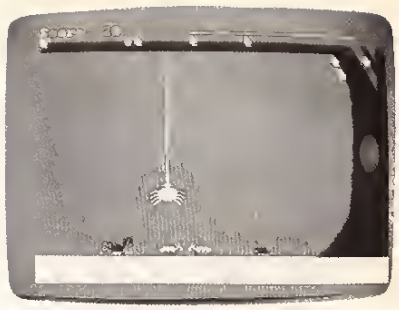
Program 3: Webster Dines Out For VIC-20

Translation by Kevin Mykytyn, Editorial Programmer

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing

```
0 PRINTCHR$(147)"VIC-20 WEBSTE
  R MAKER":PRINT        :rem 179
1 IFPEEK(43)+256*PEEK(44)=6609
  THEN3                 :rem 225
2 PRINT"POKE6608,0:POKE43,209:
  POKE44,25:NEW":PRINT"{DOWN}E
  NTER POKES, LOAD AGAIN":END
                        :rem 90
3 INPUT"{RVS}D{OFF}ISK OR
  {RVS}T{OFF}APE";D$:IFD$="D"T
  HEND=8:GOTO6          :rem 111
4 IFD$<>"T"THEN3        :rem 159
5 D=1:REM FOR 64 CHANGE 57809
  {SPACE}TO 57812 AND 63109 TO
   62957                :rem 57
6 X=0:PRINT:PRINT"READING DATA
  ..."                  :rem 107
7 READA:IFA=256THEN9    :rem 224
8 POKE4609+X,A:X=X+1:GOTO7
                        :rem 87
9 PRINT"{DOWN}SAVING WE8STER":
  IFD=8THENSYS57809"@0:WEBSTER
  ",8,0                 :rem 134
10 IFD=1THENSYS57809"WEBSTER",
  1,0                   :rem 246
11 POKE193,1:POKE194,18:POKE17
  4,65:POKE175,23:SYS63109:PR
  INT"{DOWN}OK":END     :rem 93
12 DATA 11,8,10,0,158,52,49,48
  ,57,0,0               :rem 10
13 DATA 0,169,147,32,210,255,1
  62,8,160,2,24         :rem 54
14 DATA 32,240,255,169,189,133
  ,170,169,153,160,20:rem 113
15 DATA 32,30,203,120,173,4,14
  4,9,8,141,15          :rem 253
16 DATA 144,166,170,202,208,25
  3,32,159,255,32,228:rem 112
17 DATA 255,201,13,208,233,88,
  32,197,20,169,10      :rem 216
18 DATA 141,174,2,169,25,141,1
  76,2,169,0,133        :rem 115
19 DATA 2,133,252,133,253,165,
  162,133,254,169,1     :rem 7
20 DATA 133,3,169,23,133,5,169
  ,30,133,6,160         :rem 54
21 DATA 2,169,5,153,100,3,169,
  10,153,110,3          :rem 248
22 DATA 136,16,243,169,2,141,1
  03,3,141,104,3        :rem 93
23 DATA 169,4,141,113,3,141,11
  4,3,162,4,32          :rem 250
24 DATA 17,19,202,16,250,169,1
  5,141,14,144,32       :rem 155
25 DATA 81,19,32,188,16,32,173
  ,16,32,56,17          :rem 24
26 DATA 174,176,2,32,160,18,32
  ,173,18,32,136        :rem 115
27 DATA 18,32,251,19,76,149,16
  ,165,2,208,84         :rem 81
28 DATA 165,197,201,64,240,78,
  72,32,6,17,104        :rem 124
29 DATA 201,29,208,24,166,3,22
  4,1,240,18,198        :rem 117
30 DATA 3,165,5,56,233,1,133,5
  ,165,6,233            :rem 163
31 DATA 0,133,6,76,0,17,201,37
  ,208,24,166           :rem 210
32 DATA 3,224,17,240,18,230,3,
  165,5,24,105          :rem 254
33 DATA 1,133,5,165,6,105,0,13
  3,6,76,0              :rem 55
34 DATA 17,201,32,208,8,169,1,
  133,2,169,16          :rem 12
35 DATA 133,251,169,0,32,20,17
  ,96,162,3,189         :rem 69
36 DATA 174,17,168,169,32,145,
  5,202,16,245,96       :rem 183
37 DATA 141,175,2,165,5,24,105
  ,0,133,247,165        :rem 110
38 DATA 6,105,120,133,248,162,
  3,189,174,17,168      :rem 226
39 DATA 189,178,17,145,5,173,1
  75,2,145,247,202      :rem 236
40 DATA 16,239,96,165,2,240,94
  ,32,155,17,176        :rem 129
41 DATA 3,32,6,17,165,251,16,5
  ,73,255,24            :rem 169
42 DATA 105,1,24,105,220,141,1
  2,144,198,251,48      :rem 199
43 DATA 22,160,0,169,39,145,5,
  165,5,24,105          :rem 12
44 DATA 22,133,5,165,6,105,0,1
  33,6,76,144           :rem 213
45 DATA 17,165,251,201,255,208
  ,3,32,182,17,201      :rem 206
46 DATA 239,208,9,169,0,133,2,
  141,12,144,240        :rem 110
47 DATA 13,165,5,56,233,22,133
  ,5,165,6,233          :rem 15
48 DATA 0,133,6,32,155,17,176,
  5,169,0,32            :rem 171
49 DATA 20,17,96,162,4,202,48,
  12,189,174,17         :rem 78
50 DATA 168,177,5,201,40,144,2
  43,56,96,24,96        :rem 130
51 DATA 22,23,1,0,36,35,34,33,
  169,0,141             :rem 102
52 DATA 178,2,162,4,165,3,221,
  90,3,240,40           :rem 210
53 DATA 24,105,1,221,90,3,240,
  32,202,16,238         :rem 43
54 DATA 173,178,2,201,3,144,4,
  169,100,208,6         :rem 60
```

```
55 DATA 201,2,144,13,169,50,24
   ,101,252,133,252    :rem 196
56 DATA 165,253,105,0,133,253,
   96,142,177,2,224    :rem 217
57 DATA 3,144,56,169,127,141,1
   70,2,141,12,144     :rem 161
58 DATA 32,131,17,198,251,173,
   170,2,24,105,127    :rem 214
59 DATA 141,13,144,173,170,2,3
   2,20,17,165,162     :rem 156
60 DATA 197,162,240,252,206,17
   0,2,16,230,32,6     :rem 154
61 DATA 17,206,242,31,173,242,
   31,201,48,208,3     :rem 152
62 DATA 76,101,20,96,238,178,2
   ,32,17,19,169       :rem 80
63 DATA 231,133,4,165,4,141,12
   ,144,230,252,208    :rem 198
64 DATA 2,230,253,32,251,19,23
   0,4,208,238,174     :rem 160
65 DATA 177,2,206,174,2,240,3,
   76,204,17,169       :rem 72
66 DATA 10,141,174,2,173,225,3
   1,201,54,240,41     :rem 146
67 DATA 32,71,19,74,74,74,74,1
   68,185,141,31       :rem 89
68 DATA 201,32,208,241,185,142
   ,31,201,32,208,234  :rem 45
69 DATA 169,40,153,141,31,153,
   142,31,169,0,153    :rem 213
70 DATA 141,151,153,142,151,23
   8,225,31,96,162,2   :rem 1
71 DATA 189,90,3,205,93,3,208,
   10,32,71,19         :rem 224
72 DATA 201,15,176,3,32,17,19,
   202,16,235,96       :rem 64
73 DATA 152,72,160,0,136,208,2
   53,202,208,248,104  :rem 51
74 DATA 168,96,162,4,189,90,3,
   168,222,60,3        :rem 36
75 DATA 208,66,189,70,3,157,60
   ,3,185,140,31       :rem 77
76 DATA 201,40,176,5,169,32,15
   3,140,31,189,80     :rem 170
77 DATA 3,208,23,192,22,208,12
   ,165,162,201,5      :rem 108
78 DATA 176,3,32,17,19,76,13,1
   9,254,90,3          :rem 187
79 DATA 200,76,250,18,192,0,20
   8,12,165,162,201    :rem 210
80 DATA 5,176,3,32,17,19,76,13
   ,19,222,90          :rem 177
81 DATA 3,136,185,140,31,201,4
   0,176,12,189,61     :rem 160
82 DATA 19,153,140,31,189,66,1
   9,153,140,151,202   :rem 9
83 DATA 16,159,96,173,36,145,1
   6,9,169,0,157       :rem 93
84 DATA 80,3,169,1,208,7,169,1
   ,157,80,3           :rem 132
85 DATA 169,21,157,90,3,32,71,
   19,221,100,3        :rem 11
86 DATA 144,248,221,110,3,176,
   243,157,60,3,157    :rem 219
87 DATA 70,3,96,37,37,37,38,38
   ,4,4,4              :rem 250
88 DATA 6,6,165,254,10,10,56,1
   01,254,133,254      :rem 113
89 DATA 96,169,30,141,15,144,1
   69,147,32,210,255   :rem 26
90 DATA 169,28,32,210,255,169,
   0,133,247,169,21    :rem 224
91 DATA 133,248,166,247,164,24
   8,24,32,240,255,169:rem 129
92 DATA 42,32,210,255,198,248,
   16,239,169,20,133   :rem 21
93 DATA 248,169,19,133,247,166
   ,247,164,248,24,32  :rem 88
94 DATA 240,255,169,42,32,210,
   255,198,247,16,239  :rem 79
95 DATA 230,248,165,248,201,22
   ,208,227,160,43,169:rem 119
96 DATA 5,153,162,151,169,42,1
   92,22,176,2,169     :rem 181
97 DATA 41,153,162,31,136,16,2
   37,169,40,160,0     :rem 165
98 DATA 153,140,31,169,0,153,1
   40,151,162,21,160   :rem 253
99 DATA 1,24,32,240,255,169,21
   3,160,19,32,30      :rem 113
100 DATA 203,169,43,141,159,31
   ,169,2,141,159,151  :rem 54
101 DATA 96,28,83,67,79,82,69,
   32,48,32,32         :rem 249
102 DATA 32,32,32,76,69,86,69,
   76,32,49,13         :rem 240
103 DATA 32,32,32,32,32,32,32,
   32,31,76,73         :rem 201
104 DATA 86,69,83,32,51,0,165,
   252,141,168,2       :rem 70
105 DATA 165,253,141,169,2,169
   ,0,141,170,2,141    :rem 204
106 DATA 171,2,141,172,2,141,1
   73,2,162,15,14      :rem 93
107 DATA 168,2,46,169,2,120,24
   8,173,170,2,109     :rem 167
108 DATA 170,2,141,170,2,173,1
   71,2,109,171,2      :rem 96
109 DATA 141,171,2,173,172,2,1
   09,172,2,141,172    :rem 202
110 DATA 2,216,88,202,16,216,1
   60,0,162,2,189      :rem 103
111 DATA 170,2,72,74,74,74,74,
   32,86,20,104        :rem 14
112 DATA 41,15,32,86,20,202,16
   ,236,96,205,173     :rem 157
113 DATA 2,240,9,9,48,141,173,
   2,153,213,31        :rem 3
114 DATA 200,96,104,104,162,11
   ,160,3,24,32,240    :rem 185
115 DATA 255,169,137,160,20,32
   ,30,203,32,228,255  :rem 51
116 DATA 201,89,208,3,76,62,16
   ,201,78,208,242     :rem 171
117 DATA 169,0,133,198,0,31,80
   ,76,65,89,32        :rem 30
118 DATA 65,71,65,73,78,32,89,
   47,78,0,144         :rem 252
119 DATA 87,69,66,83,84,69,82,
   32,68,73,78         :rem 19
120 DATA 69,83,32,79,85,84,13,
   13,13,13,31         :rem 225
121 DATA 32,32,72,73,84,32,82,
   69,84,85,82         :rem 240
122 DATA 78,32,84,79,32,80,76,
   65,89,0,160         :rem 244
123 DATA 0,185,0,128,153,0,28,
   185,0,129,153       :rem 55
124 DATA 0,29,136,208,241,169,
   255,141,5,144,160   :rem 6
125 DATA 95,185,231,20,153,8,2
   9,136,16,247,96     :rem 183
126 DATA 32,24,135,79,63,159,1
   59,95,8,48,193      :rem 149
127 DATA 242,252,249,249,250,2
   52,240,232,196,226,162
                       :rem 11
128 DATA 192,0,63,15,23,35,71,
   69,3,0,0            :rem 63
129 DATA 0,124,214,254,254,170
   ,170,126,129,153,231
                       :rem 150
130 DATA 219,255,36,195,1,1,1,
   1,1,1,1             :rem 253
131 DATA 1,0,60,126,255,255,25
   5,126,60,170,187    :rem 210
132 DATA 255,255,255,255,255,2
   55,255,255,255,255,255
                       :rem 28
133 DATA 255,255,255,0,0,1,3,7
   ,15,63,255,256      :rem 108
```



Multicolored bugs scurry across the lawn in "Webster Dines Out" for the IBM PC/PCjr.

Program 4: Webster Dines Out For IBM PC/PCjr

Translation by Charles Brannon, Program Editor

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
LG 1 'Webster Dines Out for IBM
     PC with Color/Graphics Adap
     tor and BASICA or Expanded
     PCjr
IG 2 PLAY "mb
EE 100 DEFINT A-Z:KEY OFF:WIDTH
     40:SCREEN 0,1,0:COLOR 7,0
     ,0:CLS:RANDOMIZE TIMER

ND 110 FOR I=1 TO 11:COLOR ,I:LO
     CATE I,1,0:PRINT SPACE$(3
     9):COLOR ,15-I:LOCATE 24-
     I,1:PRINT SPACE$(39):NEXT
FD 120 LOCATE 12,1:COLOR 0,15:PR
     INT SPACE$(12)+"Webster D
     ines Out"+SPACE$(10)

QO 125 LOCATE 16,1:PRINT"
     Do you want sound? (Y/N)"
     ;:A$=INPUT$(1):IF A$="y"
     OR A$="Y" THEN SOUNDFLAG=
     1
NJ 130 IF SOUNDFLAG THEN PLAY "l
     16o2cccd    cfffg    faaa
       ba>l8d   l16 ggged    ccc
     <ag    fffdc    l4<f
KF 135 GOSUB 550
GA 150 SCREEN 1:CLS:COLOR 1,0:LI
     NE (0,180)-(319,199),1,BF
NK 160 LINE (250,180)-(319,180),
     2:LINE (250,180)-(260,175
     ),2:LINE -(280,150),2:LIN
     E -(283,130),2:LINE -(285
     ,80),2:LINE -(283,50),2:L
     INE -(270,20),2:LINE -(0,
     20),2:: LINE (310,50)-(285
     ,13),2:LINE (310,50)-(310
     ,13),2:LINE (0,13)-(285,1
     3),2

QN 161 LINE (310,38)-(295,13),2:
     LINE (310,28)-(300,13),2:
     LINE (310,37)-(310,30),0:
     LINE (295,13)-(300,13),2:
     LINE (310,13)-(319,13),2:
     PAINT (0,14),2,2:PSET (30
     3,28),1
DI 162 A$="c113u1u1uuuuurdrdrdrd
     rdllulu":DRAW A$:PSET (27
     2,29),1:DRAW A$:PSET (278
     ,40),1:DRAW A$:FOR I=1 TO
      5:PSET (250*RND(1),15):D
     RAW A$:NEXT
```

```
AA 165 FOR I=Ø TO 15 STEP 2:CIRC
       LE (300,100),I,Ø,,,B/5:NE
       XT
DB 170 BEETSPEED=2:BEET2SPEED=2:
       BUGSPEED=2:BUG2SPEED=2:SC
       ORPSPEED=4
LC 180 BEETSCORE=10:BUGSCORE=20:
       MISSLIFE=10:BUGLIFE=10:BE
       ETLIFE=15:DRAINLIFE!=.5:P
       ROBABILITY=20
KC 190 WEBX=100:WEBY=21:PUT (WEB
       X,WEBY),WEB,PSET:LIFE!=18
       0:GOSUB 1030:LOCATE 1,1:P
       RINT"Score:";Ø
OK 200 LIFE!=LIFE!-DRAINLIFE!:SD
       UNO 40,.1*SOUNOFLAG:GOSUB
        1030
NP 210 IF SCORPFLAG THEN PUT (SC
       ORPX,SCORPY),SCORP,PSET:S
       CORPX=SCORPX-SCORPSPEED:I
       F SCORPX<Ø THEN SCORPFLAG
       =Ø:PUT (SCORPX+SCORPSPEED
       ,SCORPY),SCORP
ME 220 IF BEETFLAG THEN PUT (BEE
       TX,BEETY),BEET,PSET:BEETX
       =BEETX+BEETSPEED:IF BEETX
       >231 THEN BEETFLAG=Ø:PUT
       (BEETX-BEETSPEED,BEETY),B
       EET
KF 230 IF BEET2FLAG THEN PUT (BE
       ET2X,BEET2Y),BEET2,PSET:B
       EET2X=BEET2X-BEET2SPEED:I
       F BEET2X<Ø THEN BEET2FLAG
       =Ø:PUT (BEET2X+BEET2SPEED
       ,BEET2Y),BEET2
CH 240 IF BUGFLAG THEN PUT (BUGX
       ,BUGY),BUG,PSET:BUGX=BUGX
       -BUGSPEED:IF BUGX<Ø THEN
       BUGFLAG=Ø:PUT (BUGX+BUGSP
       EED,BUGY),BUG
BN 250 IF BUG2FLAG THEN PUT (BUG
       2X,BUG2Y),BUG2,PSET:BUG2X
       =BUG2X+BUG2SPEED:IF BUG2X
       >226 THEN BUG2FLAG=Ø:PUT
       (BUG2X-BUG2SPEED,BUG2Y),B
       UG2
AG 260 K$=INKEY$:IF K$="" THEN 4
       2Ø
FG 270 IF K$=CHR$(Ø)+CHR$(75) TH
       EN PUT (WE8X,WE8Y),WEB:WE
       BX=WEBX+10*(WEBX>Ø):PUT (
       WEBX,WEBY),WE8:GOTO 200
FE 280 IF K$=CHR$(Ø)+CHR$(77) TH
       EN PUT (WEBX,WEBY),WEB:WE
       BX=WEBX-10*(WE8X<220):PUT
        (WEBX,WE8Y),WEB:GOTO 200
ED 290 IF K$<>" " THEN 420
PL 300 FOR I=WEBY TO 145 STEP 3:
       PUT (WEBX,I),WEB,PSET
LJ 310 SOUND 500+I*20,.1*SDUNOFL
       AG
KP 320 NEXT:WBX=WEBX+15:NUMHIT=Ø
DH 330 IF BEETFLAG THEN IF ABS(W
       BX-(10+BEETX-BEETSPEED))<
       10 THEN PUT (BEETX-BEETSP
       EED,BEETY),BEET:BEETFLAG=
       Ø:LIFE!=LIFE!+BEETLIFE:SC
       ORE!=SCORE!+BEETSCORE:GOS
       UB 970:NUMHIT=NUMHIT+1
EK 340 IF BEET2FLAG THEN IF ABS(
       W8X-(10+BEET2X+BEET2SPEED
       ))<10 THEN PUT (BEET2X+BE
       ET2SPEED,BEET2Y),BEET2:BE
       ET2FLAG=Ø:LIFE!=LIFE!+BEE
       TLIFE:SCORE!=SCORE!+BEETS
       CORE:GOSUB 970:NUMHIT=NUM
       HIT+1
BI 350 IF BUGFLAG THEN IF ABS(W8
       X-(12+BUGX+BUGSPEED))<10
       THEN PUT (BUGX+BUGSPEED,B
       UGY),BUG:BUGFLAG=Ø:LIFE!=
       LIFE!+BUGLIFE:SCORE!=SCOR
       E!+BUGSCORE:GOSUB 970:NUM
       HIT=NUMHIT+1
NJ 360 IF BUG2FLAG THEN IF ABS(W
       BX-(12+BUG2X-BUG2SPEED))<
       10 THEN PUT (BUG2X-BUG2SP
       EED,BUG2Y),BUG2:BUG2FLAG=
       Ø:LIFE!=LIFE!+BUGLIFE:SCO
       RE!=SCORE!+BUGSCORE:GOSUB
        970:NUMHIT=NUMHIT+1
MN 370 IF SCORPFLAG THEN IF ABS(
       WBX-(16+SCORPX+SCDRPSPEED
       ))<10 THEN WEBY=145:GOTO
       1070
FD 380 IF NUMHIT=Ø THEN LIFE!=LI
       FE!-MISSLIFE:IF LIFE!<1 T
       HEN WEBY=145
FL 390 GOSUB 1030:IF NUMHIT>1 TH
       EN SCORE!=SCORE!+10^NUMHI
       T:GOSUB 9B0:FOR W=1 TO 50
       :SOUND 3000+10*(W AND 1),
       .05*SOUNDFLAG:NEXT:LINE (
       Ø,170)-(249,179),Ø,BF
MG 400 FOR I=140 TO WEBY STEP-3:
       PUT (WEBX,I),WEB,PSET:SOU
       ND I*20+500,.1*SOUNOFLAG:
       NEXT:PUT (WEBX,WEBY),WEB,
       PSET
CG 405 IF INKEY$<>"" THEN 405
BL 410 GOTO 200
IO 420 IF 100*RND(1)>PROBABILITY
        THEN 200
BK 430 ON 5*RND(1)+1 GOSUB 440,4
       60,4B0,500,520:GOTO 200
FC 440 IF BEETFLAG=Ø THEN BEETFL
       AG=1:BEETX=Ø
NI 450 RETURN
KA 460 IF BEET2FLAG=Ø THEN BEET2
       FLAG=1:BEET2X=231
NM 470 RETURN
FD 4B0 IF BUGFLAG=Ø THEN BUGFLAG
       =1:BUGX=226
NA 490 RETURN
CF 500 IF BUG2FLAG=Ø THEN BUG2FL
       AG=1:BUG2X=Ø
MD 510 RETURN
JL 520 IF SCORPFLAG=Ø THEN SCORP
       FLAG=1:SCDRPX=218
MF 530 RETURN
LG 540 END
LJ 550 BDTSCR=180
PK 560 READ X,Y:E=(4+INT((X+7)/B
       )*Y)/2:DIM WEB(E):WEB(Ø)=
       X:WE8(1)=Y:WEBY=BOTSCR-Y:
       FOR I=2 TO E:READ WEB(I):
       NEXT
OH 570 READ X,Y:E=(4+INT((X+7)/B
       )*Y)/2:DIM BEET(E):BEET(Ø
       )=X:BEET(1)=Y:BEETY=BOTSC
       R-Y:FOR I=2 TO E:READ BEE
       T(I):NEXT
FK 580 READ X,Y:E=(4+INT((X+7)/8
       )*Y)/2:OIM BEET2(E):BEET2
       (Ø)=X:BEET2(1)=Y:BEET2Y=B
       DTSCR-Y:FDR I=2 TO E:READ
        8EET2(I):NEXT
HC 590 READ X,Y:E=(4+INT((X+7)/B
       )*Y)/2:DIM BUG(E):8UG(Ø)=
       X:BUG(1)=Y:BUGY=8OTSCR-Y:
       FDR I=2 TO E:READ BUG(I):
       NEXT
EA 600 READ X,Y:E=(4+INT((X+7)/8
       )*Y)/2:OIM BUG2(E):8UG2(Ø
       )=X:BUG2(1)=Y:BUG2Y=8DTSC
       R-Y:FDR I=2 TO E:READ BUG
       2(I):NEXT
CD 610 READ X,Y:E=(4+1NT((X+7)/B
       )*Y)/2:DIM SCDRP(E):SCDRP
       (Ø)=X:SCDRP(1)=Y:SCDRPY=8
       DTSCR-Y:FOR I=2 TO E:READ
        SCORP(I):NEXT
ME 620 RETURN
LF 630 ENO
JO 640 DATA &H3C,&H1a,&HØ,&H500,
       &HØ,&HØ,&HØ,&H500
EI 650 DATA &HØ,&HØ,&HØ,&H500,&H
       Ø,&HØ,&HØ,&HFFØØ
CG 660 DATA &HFØ,&HØ,&HØ,&HFFØF,
       &HFF,&HØ,&HØ,&HBFØA
HE 670 DATA &HEA,&HØ,&HØ,&HBFØA,
       &HEA,&HØ,&HØ,&HBFØ2
IB 6BØ DATA &HEB,&HØ,&HØ,&H3FØØ,
       &HCØ,&HØ,&HØ,&HFFØF
GL 690 DATA &HCØFF,&HØ,&HØ,&HFFF
       F,&HFCFF,&HØ,&H57Ø1,&HF50
       F
HC 700 DATA &H7OFF,&H54,&HF14,&H
       O5F5,&HFF75,&H4ØØ1,&HF4Ø,
       &H75FF
JP 710 DATA &HFFOF,&H1ØØØ,&H5FØ1
       ,&H55F7,&HFF5O,&H54,&H314
       ,&HF55F
NO 720 DATA &H5CFF,&H4ØØ1,&H154Ø
       ,&HO5FF,&HF57F,&H1Ø4Ø,&H4
       ØØ1,&HD5FF
EL 730 DATA &HFØ7F,&H14,&H1Ø4,&H
       F543,&H14FC,&H1,&H141Ø,&H
       Ø
BI 740 DATA &H1ØØ,&H4Ø4Ø,&H4ØØØ,
       &HØ,&HØ,&H1Ø,&H1,&HØ
QL 750 DATA &HØ,&H4,&HØ,&HØ,&HØ,
       &HØ,&HØ,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
       ,Ø,Ø,Ø
PP 760 DATA &H26,&H5,&HØ,&HABAA,
       &HAØ,&HAA2A,&H5BAA,&HAAØØ
LI 770 DATA &HABAA,&HAØ,&H1101,&
       H1Ø,&HØ,&HØ,&HØ,&HFFØØ
EL 7BØ DATA &H24,&H5,&HAA2B,&HAB
       ,&H96ØØ,&HAAAA,&H8Ø,&HAA2
       B
IJ 790 DATA &HAØAA,&HØ,&H411Ø,&H
       Ø,&HØ,&HØ,&HØ,&HØ
MP BØØ DATA &H3Ø,&HA,&H3,&HØ,&HØ
       ,&H3ØØC,&HØ,&HØ
NI B1Ø DATA &HCØØ5,&HØ,&HØ,&H439
       5,&HFFFF,&HCØ,&H5F5A,&HFF
       FF
FI B2Ø DATA &HF4,&H55A,&H5555,&H
       55,&H9ØØ,&H5A5A,&H54,&H2Ø
       ØØ
IO 83Ø DATA &H2Ø2Ø,&HØ,&H2ØØØ,&H
       2Ø2Ø,&HØ,&HØ,&HØ,&HØ
NC 84Ø DATA &HØ
MB 85Ø DATA &H3Ø,&HA,&HØ,&HØ,&HC
       ØØØ,&HØ,&HØ,&H3ØØC
OA 86Ø DATA &HØ,&HØ,&H5ØØ3,&H3ØØ
       ,&HFFFF,&H56C1,&H1FØØ,&HF
       FFF
LO 87Ø DATA &HA5F5,&H55ØØ,&H5555
       ,&HA55Ø,&H15ØØ,&HA5A5,&H6
       Ø,&HØ
KG 88Ø DATA &HBØ8,&H8,&HØ,&HBØB,
       &H8,&HØ,&HØ,&HØ
OM 89Ø DATA &HØ
LL 9ØØ DATA &H4Ø,&HD,&HØ,&HØ,&H2
       AØØ,&HB8Ø,&HØ,&HØ
ON 91Ø DATA &HA2Ø2,&HAØ,&HØ,&HØ,
       &H82Ø2,&HAØ,&HBØC,&HØ
MD 92Ø DATA &H2ØA,&H8Ø,&H2633,&H
       Ø,&HA,&HCØ,&HE6ØØ,&HA882
BP 93Ø DATA &H32A,&HØ,&HA63Ø,&HA
       AAA,&HA8,&HØ,&HAAC3,&HAAA
       A
KO 94Ø DATA &HAØ,&HØ,&H2A3C,&HAA
       2A,&H8Ø,&HØ,&HØ,&HAAØA
PF 95Ø DATA &H8Ø,&HØ,&HØ,&H1111,
       &HØ,&HØ,&HØ,&H4444
CL 96Ø DATA &HØ,&HØ,&HØ,&HØ,&HØ,
       &HØ,&HF4Ø
IN 97Ø LOCATE 1,1:PRINT"Yummy!":
       SDUNO 11Ø,2*SDUNDFLAG:FDR
        I=6ØØ TO 5ØØ STEP-2:SDUN
       O I,.1*SOUNOFLAG:NEXT I
JE 98Ø LOCATE 1,1:PRINT"Score:";
       SCDRE!
```

```
JB 990 IF SCORE'>500 THEN BUGSPE
       ED=3:BUG2SPEED=3:SCORPSPE
       ED=5:PROBABILITY=18:DRAIN
       LIFE'=.6:MISSLIFE=15
FA 1000 IF SCORE'>1000 THEN BUGS
       PEED=4:BUG2SPEED=4:BEETS
       PEED=3:BEET2SPEED=3:SCOR
       PSPEED=6:PROBABILITY=17:
       DRAINLIFE'=1
LL 1010 IF SCORE'>5000 THEN BEET
       SPEED=4:BEET2SPEED=4:DRA
       INLIFE'=2:PROBABILITY=15
       :MISSLIFE=20
IB 1020 RETURN
JG 1030 IF LIFE'>180 THEN LIFE'=
       180
IB 1040 LINE (136,5)-(136+LIFE!,
       5),-2*(LIFE'<30)-3*(LIFE
       '>30)+2*(LIFE'>=60):LINE
        (137+LIFE',5)-(319,5),0
OI 1050 IF LIFE'<1 THEN 1110
JN 1060 RETURN
EA 1070 FOR J=1 TO 5
IJ 1080 FOR I=145 TO 130 STEP-3:
       PUT (WEBX,I),WEB,PSET:NE
       XT:SOUND 50,SOUNDFLAG
IG 1090 FOR I=130 TO 145 STEP 3:
       PUT (WEBX,I),WEB,PSET:NE
       XT:SOUND 60,SOUNDFLAG
GM 1100 NEXT J
AF 1110 FOR I=1 TO 10:PUT (WEBX+
       5*RND(1)+5*RND(1)*(WEBX>
       5),WEBY+5*RND(1)-5*RND(1
       )),WEB:SOUND 40,.1*SOUND
       FLAG:NEXT
MD 1120 IF INKEY$<>"" THEN 1120
JL 1130 LOCATE 13,13:PRINT"Play
       Again? (Y/N)";:A$=INPUT$
       (1):IF A$="y" OR A$="Y"
       THEN RUN
HH 1140 SCREEN 0,0,0
HC 1150 END
```

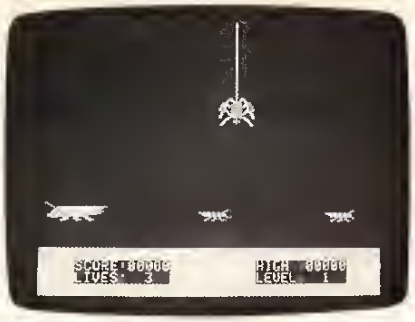
Program 5: Webster Dines Out For Apple

Translation by Tim Victor, Editorial Programmer

Please refer to the article "Apple MLX" elsewhere in this issue before entering this listing.

```
START ADDRESS: 1100
END ADDRESS:   1F14
1100: 20 AC 19 A9 00 8D 0D 08 73
1108: 8D 0E 08 A9 20 8D 1E 08 88
1110: 85 E6 20 F2 F3 2C 57 C0 A1
1118: 2C 52 C0 2C 50 C0 2C 10 AD
1120: C0 A9 40 85 E6 20 F2 F3 FE
1128: A9 60 85 E6 20 F2 F3 A9 84
1130: 50 85 FC A9 06 85 FD A9 02
1138: 20 A0 27 91 FC 88 10 F8 86
1140: E6 FD A5 FD C9 08 D0 04 FD
1148: A9 0A 85 FD C9 0C D0 E7 5A
1150: A5 FC 69 7F 90 D8 2C 53 49
1158: C0 A0 04 A9 01 20 78 19 B0
1160: A0 70 A9 1A A2 08 20 98 DF
1168: 19 A0 04 A9 02 20 78 19 F4
1170: A0 7B A9 1A A2 08 20 98 B2
1178: 19 A0 19 A9 01 20 78 19 9F
1180: A0 86 A9 1A A2 0B 20 98 85
1188: 19 A0 19 A9 02 20 78 19 87
1190: A0 91 A9 1A A2 0B 20 98 58
1198: 19 A9 00 A8 99 88 08 C8 C4
11A0: C0 78 D0 F8 20 A3 16 20 C6
11A8: 04 17 20 5A 16 A9 00 8D 21
11B0: 18 08 8D 19 08 2C 2E 08 79
11B8: 30 06 20 60 12 4C F6 11 3F
11C0: 20 25 13 B0 26 2C 2F 08 F1
11C8: 10 13 20 3B 14 90 08 20 82
11D0: 58 14 90 17 4C 0B 11 20 78
11D8: E9 14 20 A7 15 2C 2E 08 31
11E0: 10 09 20 39 15 20 72 16 09
11E8: 4C F6 11 A9 00 8D 21 09 2D
11F0: 8D 28 09 8D 2E 08 20 DD 8D
11F8: 12 AD 18 08 F0 17 20 9F D6
1200: 16 AD 18 08 18 6D 11 08 8E
1208: 8D 11 08 90 08 20 4F 13 83
1210: 90 03 4C 08 11 AD 19 08 F0
1218: F0 03 20 98 16 AD 00 C0 58
1220: 10 38 2C 10 C0 C9 8D D0 8A
1228: 03 20 98 15 2C 2E 08 30 F4
1230: 2C C9 88 D0 0A AD 27 09 59
1238: F0 23 CE 27 09 10 1E C9 78
1240: 95 D0 0C AD 27 09 C9 23 D3
1248: F0 13 EE 27 09 10 0E C9 68
1250: A0 D0 0A A9 80 8D 2E 08 73
1258: A9 08 8D 30 08 4C A7 11 D9
1260: A2 00 A0 00 8C 20 08 8D 9C
1268: 08 09 D0 29 20 6F 15 29 91
1270: DE D0 5C 20 6F 15 10 08 C0
1278: A9 01 99 32 08 A0 85 A9 DF
1280: 1A D0 09 A9 FF 99 32 08 74
1288: A0 8A A9 1A 20 7C 15 EE 8E
1290: 18 08 4C CF 12 89 32 08 2D
1298: 2C 1E 08 F0 03 70 12 2C 94
12A0: 50 0F 18 7D 09 09 9D 09 3C
12A8: 09 C9 FD 30 1D C9 28 10 F6
12B0: 19 38 8D 09 09 ED 2C 09 19
12B8: 90 15 C9 03 B0 11 AD 28 24
12C0: 09 F0 0C 20 6F 15 29 03 4E
12C8: D0 05 A9 00 9D 08 09 8A 75
12D0: 18 69 05 AA AC 20 08 C8 65
12D8: C0 04 D0 88 60 AD 28 09 1A
12E0: D0 2B 2C 2E 08 30 3D 20 3C
12E8: 6F 15 29 DC D0 36 A9 00 AF
12F0: 38 ED 31 08 8D 31 08 30 C4
12F8: 06 A0 A8 A9 1A D0 04 A0 15
1300: B0 A9 1A A2 23 20 7C 15 FD
1308: EE 19 08 D0 17 AD 2C 09 CA
1310: 18 6D 31 08 8D 2C 09 C9 3D
1318: F9 30 04 C9 28 30 05 A9 1A
1320: 00 8D 28 09 60 AD 28 09 82
1328: D0 08 AD 30 08 10 03 38 30
1330: 80 1C C9 78 90 08 A0 F8 55
1338: 8C 30 08 A0 80 2C A0 00 81
1340: 8C 2F 08 18 AD 28 09 6D 88
1348: 30 08 8D 28 09 18 60 A9 CF
1350: D0 8D 11 08 AC 10 08 C0 58
1358: 86 D0 03 4C 81 14 C8 8C 2F
1360: 10 08 20 49 16 A9 28 8D 5E
1368: 17 08 A9 47 8D 17 18 A9 68
1370: 18 8D 18 18 A9 14 85 FC 30
1378: A9 18 85 FD 20 6F 15 8D 41
1380: 16 08 C9 A0 80 F6 4A 4A 37
1388: 29 FE 8D 07 08 AD 16 08 50
1390: 29 07 C9 07 F0 E6 18 69 73
1398: 02 8D 16 08 29 01 F0 06 98
13A0: EE 07 08 A0 06 2C A0 02 2F
13A8: 81 FC 85 EE C8 81 FC 85 13
13B0: EF C8 81 FC 85 FE C8 81 72
13B8: FC 85 FF A9 02 8D 12 08 C8
13C0: A9 00 8D 13 08 A9 78 8D 04
13C8: 14 08 A9 7F 8D 15 08 AC A5
13D0: 16 08 F0 24 0E 12 08 10 3D
13D8: 08 AD 12 08 69 80 8D 12 A8
13E0: 08 2E 13 08 38 2E 14 08 24
13E8: 10 08 AD 14 08 69 80 8D 84
13F0: 14 08 2E 15 08 88 D0 DC 18
13F8: A0 08 AD 13 08 91 EE AD 2B
1400: 15 08 91 FE 88 AD 12 08 FD
1408: 91 EE AD 14 08 91 FE 88 88
1410: 10 E8 A9 87 8D 08 08 20 E4
1418: DC 17 AD 10 08 C9 B4 90 8C
1420: 10 A9 14 85 FC A9 1B 85 DF
1428: FD A9 81 8D 08 08 20 DC 40
1430: 17 CE 17 08 F0 03 4C 74 98
1438: 13 18 60 AD 27 09 2C 31 BD
1440: 08 30 03 38 E9 07 CD 2C 8F
1448: 09 F0 0C 90 03 18 90 07 5C
1450: 69 04 CD 2C 09 90 00 60 95
1458: A9 00 8D 21 09 A9 78 8D 86
1460: 28 09 18 AD 2C 09 6D 31 4E
1468: 08 8D 2C 09 C9 F9 30 28 CF
1470: C9 28 10 27 2C 31 08 30 62
1478: 03 A9 05 2C A9 FE 18 6D D6
1480: 2C 09 8D 27 09 C9 FC 30 BE
1488: 04 C9 28 30 05 A9 00 8D 89
1490: 26 09 20 90 16 20 04 17 6B
1498: 4C 58 14 A0 C9 A9 1A A2 55
14A0: 1E 20 7C 15 CE 0F 08 20 A3
14A8: 38 16 AD 0F 08 C9 80 D0 82
14B0: 36 A0 0D A9 03 20 78 19 FA
14B8: A0 9C A9 1A A2 0F 20 98 59
14C0: 19 2C 10 C0 AD 00 C0 10 8D
14C8: FB 2C 10 C0 C9 CE F0 04 77
14D0: C9 EE D0 03 4C F1 16 A9 E3
14D8: 20 A0 00 99 D0 07 99 D0 79
14E0: 08 C8 C0 28 D0 F5 60 18 92
14E8: 60 A2 00 BE 21 08 8D 08 7F
14F0: 09 F0 3C BD 09 09 38 E9 04
14F8: 04 CD 27 09 10 31 18 69 EA
1500: 06 CD 27 09 30 29 A9 00 8F
1508: 9D 08 09 AD 21 08 F0 0E 18
1510: 18 AD 0B 08 69 32 8D 08 CD
1518: 08 90 03 EE 0C 08 EE 21 39
1520: 08 18 AD 08 08 69 19 8D 60
1528: 08 08 90 03 EE 0C 08 8A 5E
1530: 18 69 05 AA C9 14 D0 86 03
1538: 60 AD 1E 08 29 20 F0 0C F9
1540: AD 28 09 8D 00 18 A0 8F 82
1548: A9 1A D0 0A AD 28 09 8D 36
1550: 0A 18 A0 C4 A9 1A A2 19 8A
1558: 20 7C 15 AD 27 09 18 69 26
1560: 02 8D 22 09 AD 28 09 F0 D4
1568: 02 A9 01 8D 21 09 60 AD 92
1570: 23 08 0A 0A 38 6D 23 08 D5
1578: 8D 23 08 60 8C 8A 15 8D 7F
1580: 88 15 A0 04 8A 18 69 04 95
1588: AA 89 FF FF 9D 08 09 CA 60
1590: 88 10 F6 E8 AC 8A 15 AD D7
1598: 88 15 60 2C 10 C0 2C 00 78
15A0: C0 10 FB 2C 10 C0 60 20 D5
15A8: C5 15 AE 08 08 EC 0D 08 97
15B0: AD 0C 08 ED 0E 08 90 0C 52
15B8: 8E 0D 08 AD 0C 08 8D 0E F2
15C0: 08 20 D8 15 60 AC 08 08 37
15C8: AD 0C 08 20 F2 15 A0 0A 07
15D0: A9 01 20 78 19 4C E8 15 7C
15D8: AC 0D 08 AD 0E 08 20 F2 3C
15E0: 15 A0 1F A9 01 20 78 19 CE
15E8: A0 29 A9 08 A2 05 20 98 65
15F0: 19 60 8C 27 08 8D 28 08 92
15F8: A2 04 A9 80 9D 29 08 CA 22
1600: 10 FA E8 AD 27 08 DD 34 34
1608: 16 AD 28 08 FD 30 16 90 9D
1610: 11 8D 28 08 AD 27 08 FD C5
1618: 34 16 8D 27 08 FE 29 08 9E
1620: D0 E1 E8 E0 04 D0 DC AD 23
1628: 27 08 69 AF 9D 29 08 60 14
1630: 27 03 00 00 10 E8 64 0A A7
1638: A0 0C A9 02 20 78 19 A0 C2
1640: 0F A9 08 A2 01 20 98 19 5C
1648: 60 A0 21 A9 02 20 78 19 26
1650: A0 10 A9 08 A2 01 20 98 78
1658: 19 60 A9 80 8D 68 1A A9 5E
1660: 07 8D 0A 08 8D 09 08 A9 7F
1668: 39 8D 6D 1A A9 08 8D 6E DA
1670: 1A 60 AD 28 09 4A 4A 4A 4A
1678: 69 06 8D 0A 08 8D 09 08 8D
1680: A9 80 8D 68 1A A9 38 8D 7F
1688: 6D 1A A9 08 8D 6E 1A 60 62
1690: AD 1E 08 29 20 F0 C3 A9 A4
1698: 0A D0 DF A9 24 D0 D8 A9 5A
16A0: 06 D0 D7 A9 83 8D 0F 08 93
16A8: 20 38 16 A9 81 8D 10 08 3C
16B0: A9 D0 8D 11 08 20 49 16 12
16B8: A9 00 8D 08 08 8D 0C 08 82
16C0: 20 C5 15 20 D8 15 A0 00 6F
16C8: 98 99 08 09 C8 C0 28 D0 A3
16D0: F8 A9 00 A0 7F 99 38 08 C8
16D8: 88 D0 FA A9 80 8D 38 08 2A
16E0: 20 5A 16 A9 01 8D 31 08 89
16E8: A0 C9 A9 1A A2 1E 4C 7C 51
16F0: 15 2C 54 C0 2C 51 C0 AD 1F
16F8: 00 8E C9 4C F0 03 4C D0 CF
1700: 03 4C 00 8E AD 1E 08 49 ED
1708: 60 8D 1E 08 29 20 F0 02 88
1710: A9 01 AA 8D 54 C0 A9 38 85
1718: 8D 17 18 A9 18 8D 18 18 AF
1720: A9 00 8D 1F 08 8D 22 08 89
1728: 0A 0A 6D 22 08 8D 22 08 70
1730: AD 1E 08 29 40 F0 02 A9 C3
```

```
1738: 28 18 6D 22 08 AA 8D 88 6F
1740: 08 F0 1E 8D 88 08 85 FC 54
1748: 8D 8C 08 85 FD 8D 89 08 40
1750: 8D 07 08 8D 8A 08 8D 08 FC
1758: 08 20 DC 17 A9 00 9D 88 E0
1760: 08 20 8C 18 D0 8F A9 1F A5
1768: 8D 17 18 A9 18 8D 18 18 FF
1770: A9 00 8D 1F 08 8D 22 08 D9
1778: 0A 0A 6D 22 08 AA 8D 08 6C
1780: 09 D0 03 4C D6 17 8D 08 26
1788: 09 85 FC 8D 0C 09 85 FD A5
1790: 8D 09 09 8D 07 08 8D 0A 87
1798: 09 8D 08 08 AD 1F 08 8D 87
17A0: 22 08 0A 0A 6D 22 08 8D 55
17A8: 22 08 AD 1E 08 29 40 F0 D7
17B0: 02 A9 28 18 6D 22 08 AA 7F
17B8: A9 01 9D 88 08 A5 FC 9D A9
17C0: 88 08 A5 FD 9D 8C 08 AD 01
17C8: 07 08 9D 89 08 AD 08 08 DA
17D0: 9D 8A 08 20 DC 17 20 8C 8F
17D8: 18 D0 9A 60 A0 00 81 FC 06
17E0: 8D 18 08 C8 81 FC 8D 1C E2
17E8: 08 8D 1D 08 AD 08 08 8D CD
17F0: 1A 08 AD 07 08 29 01 0A 45
17F8: 0A 69 02 A8 81 FC 85 EE CC
1800: C8 81 FC 85 EF C8 81 FC F8
1808: 85 FE C8 81 FC 85 FF 20 0D
1810: CF 18 AC 1D 08 88 20 FF 38
1818: FF 20 6D 18 D0 F1 60 81 40
1820: EC 11 FE 31 FC 8D 2F 18 92
1828: 81 EC 49 7F 31 EE 09 00 E4
1830: 91 FC 20 95 18 88 10 E7 80
1838: 81 EC 11 FE 31 FC 91 FC 2C
1840: 20 95 18 88 10 F2 60 A5 24
1848: FC 8D 5C 18 A5 FD 49 60 7F
1850: 8D 5D 18 81 FC 31 EC 11 54
1858: EE 91 FC 99 FF FF 81 FE FF
1860: 49 7F 11 EC 91 EC 20 95 1C
1868: 18 88 10 E7 60 18 AD 1C 22
1870: 08 65 EE 85 EE 90 02 E6 D8
1878: EF 18 AD 1C 08 65 FE 85 77
1880: FE 90 02 E6 FF EE 1A 08 FA
1888: CE 18 08 60 EE 1F 08 AD 9F
1890: 1F 08 C9 08 60 20 6A 1A 7E
1898: F0 03 2C 30 C0 4A 90 0A E4
18A0: EE 6D 1A D0 03 EE 6E 1A 8E
18A8: A9 80 8D 68 1A CE 09 08 5C
18B0: D0 1C A9 80 8D 68 1A AD 89
18B8: 0A 08 8D 09 08 4A 4A 4A 7A
18C0: 38 49 FF 6D 6D 1A 8D 6D 92
18C8: 1A 80 03 CE 6E 1A 60 AD C9
18D0: 07 08 10 27 A5 EE 38 ED 42
18D8: 07 08 85 EE 90 02 E6 EF 78
18E0: A5 FE 38 ED 07 08 85 FE E8
18E8: 90 02 E6 FF AD 1D 08 18 C8
18F0: 6D 07 08 8D 1D 08 A9 00 CF
18F8: 8D 07 08 38 A9 28 ED 07 07
1900: 08 CD 1D 08 80 03 8D 1D 97
1908: 08 AD 1A 08 29 3F A8 89 8E
1910: 38 19 0D 1E 08 85 FD 09 83
1918: 60 85 ED AD 1A 08 29 08 8F
1920: C9 08 A9 00 6A 2C 1A 08 AE
1928: 10 02 69 50 50 02 69 28 9A
1930: 6D 07 08 85 FC 85 EC 60 6C
1938: 00 04 08 0C 10 14 18 1C 4A
1940: 00 04 08 0C 10 14 18 1C 52
1948: 01 05 09 0D 11 15 19 1D 5A
1950: 01 05 09 0D 11 15 19 1D 62
1958: 02 06 0A 0E 12 16 1A 1E 6A
1960: 02 06 0A 0E 12 16 1A 1E 72
1968: 03 07 08 0F 13 17 18 1F 7A
1970: 03 07 08 0F 13 17 18 1F 82
1978: 8C A3 19 18 6A 6A 08 18 A3
1980: 69 50 6D A3 19 8D A3 19 C2
1988: 8D A6 19 28 A9 03 2A 8D 0C
1990: A4 19 69 04 8D A7 19 60 66
1998: CA 8C A0 19 8D A1 19 8D D8
19A0: FF FF 9D FF FF 9D FF FF FC
19A8: CA 10 F4 60 A9 CE 85 FC 79
19B0: A9 1A 85 FD A9 00 85 FE 26
19B8: A9 0C 85 FF A9 08 8D 26 22
19C0: 08 A0 00 81 FC 8D 18 08 96
19C8: C8 81 FC 8D 1C 08 C8 A5 7C
19D0: FE 91 FC C8 A5 FF 91 FC 60
19D8: A0 06 81 FC 85 EE C8 81 0E
19E0: FC 85 EF AC 1C 08 88 81 7F
19E8: EE 4A 91 FE 88 30 03 20 72
19F0: 5D 1A 20 44 1A CE 18 08 EA
19F8: D0 E9 A0 00 81 FC 8D 18 D9
1A00: 08 A0 04 A5 FE 91 FC C8 3C
1A08: A5 FF 91 FC A0 08 81 FC 96
1A10: 85 EE C8 81 FC 85 EF AC 81
1A18: 1C 08 88 81 EE 09 80 4A 6F
1A20: 91 FE 88 30 03 20 5D 1A 5E
1A28: 20 44 1A CE 18 08 D0 E7 30
1A30: 18 A5 FC 69 0A 85 FC 90 01
1A38: 02 E6 FD CE 26 08 F0 03 0A
1A40: 4C C1 19 60 18 AD 1C 08 E8
1A48: 65 EE 85 EE 90 02 E6 EF D4
1A50: 18 AD 1C 08 65 FE 85 FE 31
1A58: 90 02 E6 FF 60 81 EE 90 6A
1A60: 02 09 80 4A 91 FE 88 10 36
1A68: F4 60 A9 00 2C FF FF 60 26
1A70: D3 C3 CF D2 C5 8A A0 A0 A1
1A78: A0 A0 A0 CC C9 D6 C5 D3 0F
1A80: 8A A0 A0 A0 A0 A0 C8 C9 38
1A88: C7 C8 A0 8A A0 A0 A0 A0 F8
1A90: A0 CC C5 D6 C5 CC 8A A0 E5
1A98: A0 A0 A0 A0 D0 D2 C5 D3 94
1AA0: D3 A0 CE A0 D4 CF A0 D1 C3
1AA8: D5 C9 D4 01 F9 80 CE 1A 6E
1AB0: 01 27 80 D8 1A 01 FD 85 23
1AB8: E2 1A 01 27 85 EC 1A 01 8C
1AC0: 00 04 00 18 01 00 04 0A C1
1AC8: 18 01 12 00 F6 1A 0D 08 4F
1AD0: 00 00 00 00 1E 18 86 18 8A
1AD8: 0D 08 00 00 00 00 EE 18 8E
1AE0: 56 1C 08 04 00 00 00 00 88
1AE8: 8E 1C DE 1C 08 04 00 00 71
1AF0: 00 00 FE 1C 1E 1D 13 05 57
1AF8: 00 00 00 00 3E 1D 9D 1D E8
1B00: 80 01 00 00 00 00 FC 1D CD
1B08: 7C 1E 80 01 00 00 00 00 24
1B10: FC 1D 7C 1E 06 02 00 00 85
1B18: 00 00 FC 1E 08 1F 00 00 8C
1B20: 00 00 00 00 00 08 00 00 76
1B28: 00 00 00 00 00 02 00 00 66
1B30: 00 00 00 00 40 00 60 7F A8
1B38: 7F 7F 7F 7F 13 00 00 7E 0D
1B40: 7F 7F 7F 7F 05 00 00 78 9E
1B48: 7F 7F 7F 7F 1D 00 00 2A 19
1B50: 7F 7F 7F 3F 3D 00 00 28 1C
1B58: 75 7F 7F 3F 7D 00 00 00 F8
1B60: 55 7E 7F 2F 75 02 00 00 77
1B68: 40 2A 55 2A 55 02 00 00 49
1B70: 40 02 05 0A 54 00 00 00 28
1B78: 50 00 05 0A 00 00 00 00 18
1B80: 14 20 01 28 00 00 7F 7F E9
1B88: 7F 7F 7F 7F 7F 63 7F 7F 4E
1B90: 7F 7F 7F 7F 7F 78 7F 7F AA
1B98: 7F 7F 7F 7F 1F 7E 0F 00 67
1BA0: 00 00 00 00 40 7F 7F 00 D5
1BA8: 00 00 00 00 70 7F 7F 03 62
1BB0: 00 00 00 00 40 7F 7F 00 E5
1BB8: 00 00 00 00 00 7F 7F 03 EE
1BC0: 00 00 00 00 00 7E 7F 3F 2F
1BC8: 00 00 00 00 00 78 7F 7F 5F
1BD0: 1F 00 00 00 00 78 7F 7F F6
1BD8: 1F 38 70 60 01 7E 7F 7F 41
1BE0: 07 3E 70 60 7E 7E 7F 7F AA
1BE8: 41 0F 7C 03 7F 7F 40 00 8D
1BF0: 00 00 00 00 00 00 00 02 29
1BF8: 00 00 00 00 00 00 00 08 37
1C00: 00 00 00 00 00 00 00 20 58
1C08: 7E 7F 7F 7F 7F 1F 00 00 8F
1C10: 7D 7F 7F 7F 7F 03 00 60 37
1C18: 7D 7F 7F 7F 7F 00 00 70 43
1C20: 75 7F 7F 7F 57 02 00 78 16
1C28: 75 7F 7F 3F 55 00 00 3A C3
1C30: 55 7F 7F 28 05 00 00 2A E7
1C38: 55 2A 55 0A 00 00 00 28 19
1C40: 41 02 05 0A 00 00 00 00 DA
1C48: 40 02 05 28 00 00 00 00 44
1C50: 50 00 14 20 01 00 1F 7E F9
1C58: 7F 7F 7F 7F 7F 7F 7F 78 89
1C60: 7F 7F 7F 7F 7F 7F 7F 63 7C
1C68: 7F 7F 7F 7F 7F 7F 7F 0F 30
1C70: 00 00 00 00 00 40 7F 3F E7
1C78: 00 00 00 00 00 78 7F 0F A0
1C80: 00 00 00 00 00 7E 7F 07 88
1C88: 00 00 00 00 00 78 7F 03 A4
1C90: 00 00 00 00 00 7E 7F 00 C1
1C98: 00 00 00 00 70 7F 7F 00 51
1CA0: 00 00 00 60 7F 7F 7F 03 DA
1CA8: 18 38 70 60 7F 7F 7F 7F 87
1CB0: 1F 38 70 03 7F 7F 7F 7F 3D
1CB8: 07 7E 41 0F 7C 7F 00 00 0F
1CC0: 00 20 00 00 00 08 58 38 0D
1CC8: 77 06 40 38 77 0E 00 3A 28
1CD0: 77 0E 00 20 44 20 00 08 F4
1CD8: 44 00 00 02 01 02 7F 7F E1
1CE0: 7F 0F 7F 7F 7F 63 03 00 14
1CE8: 00 70 0F 00 00 60 7F 00 9F
1CF0: 00 60 7F 0F 11 0E 7F 63 45
1CF8: 11 7F 7F 38 7C 78 04 00 DA
1D00: 00 00 10 00 00 00 60 6E 68
1D08: 5D 18 70 6E 5D 03 70 6E F2
1D10: 5D 00 04 22 04 00 00 22 DD
1D18: 10 00 40 00 41 00 71 7F CE
1D20: 7F 7F 47 7F 7F 7F 0F 00 F2
1D28: 00 40 07 00 00 78 07 00 43
1D30: 00 7E 71 08 71 7F 7F 08 49
1D38: 47 7F 1F 3F 1C 7E 00 40 E8
1D40: 03 1C 00 00 60 06 36 00 8A
1D48: 00 30 0C 63 00 00 1E 38 8A
1D50: 43 07 00 38 6E 65 0D 40 05
1D58: 6D 3A 37 18 40 4D 6E 15 99
1D60: 18 60 0C 3A 07 33 60 1E 49
1D68: 6E 45 37 00 37 3A 67 0E 91
1D70: 40 61 6E 35 18 40 41 3A C2
1D78: 17 18 60 00 6E 05 30 60 98
1D80: 60 39 38 30 00 38 20 60 25
1D88: 01 00 0C 28 01 03 00 06 61
1D90: 28 01 06 00 03 20 00 0C 84
1D98: 40 01 08 01 18 7F 1F 78 89
1DA0: 41 7F 7F 0F 70 00 7F 7F 3E
1DA8: 07 21 08 7E 7F 40 03 18 82
1DB0: 70 3F 00 00 00 60 1F 00 82
1DB8: 00 00 40 1F 00 00 00 40 2D
1DC0: 0F 60 00 30 00 0F 40 00 5A
1DC8: 10 00 3F 00 00 00 60 1F D2
1DD0: 0C 00 00 43 1F 1C 00 40 EE
1DD8: 43 0F 7E 00 70 07 0F 0E 14
1DE0: 00 00 07 7F 03 0F 0F 7C E2
1DE8: 7F 61 03 3C 78 7F 70 03 05
1DF0: 7C 70 3F 78 0F 7F 61 1F 4D
1DF8: 7C 03 7C 43 20 20 20 20 D7
1E00: 20 20 20 20 20 20 20 20 3C
1E08: 20 20 20 20 20 20 20 20 44
1E10: 20 20 20 20 20 20 20 20 4C
1E18: 20 20 20 20 20 20 20 20 54
1E20: 20 20 20 20 20 20 20 20 5C
1E28: 20 20 20 20 20 20 20 20 64
1E30: 20 20 20 20 20 20 20 20 6C
1E38: 20 20 20 20 20 20 20 20 74
1E40: 20 20 20 20 20 20 20 20 7C
1E48: 20 20 20 20 20 20 20 20 84
1E50: 20 20 20 20 20 20 20 20 8C
1E58: 20 20 20 20 20 20 20 20 94
1E60: 20 20 20 20 20 20 20 20 9C
1E68: 20 20 20 20 20 20 20 20 A4
1E70: 20 20 20 20 20 20 20 20 AC
1E78: 20 20 20 20 0F 0F 0F 0F 84
1E80: 0F 0F 0F 0F 0F 0F 0F 0F 8C
1E88: 0F 0F 0F 0F 0F 0F 0F 0F C4
1E90: 0F 0F 0F 0F 0F 0F 0F 0F CC
1E98: 0F 0F 0F 0F 0F 0F 0F 0F D4
1EA0: 0F 0F 0F 0F 0F 0F 0F 0F DC
1EA8: 0F 0F 0F 0F 0F 0F 0F 0F E4
1EB0: 0F 0F 0F 0F 0F 0F 0F 0F EC
1EB8: 0F 0F 0F 0F 0F 0F 0F 0F F4
1EC0: 0F 0F 0F 0F 0F 0F 0F 0F FC
1EC8: 0F 0F 0F 0F 0F 0F 0F 0F 05
1ED0: 0F 0F 0F 0F 0F 0F 0F 0F 0D
1ED8: 0F 0F 0F 0F 0F 0F 0F 0F 15
1EE0: 0F 0F 0F 0F 0F 0F 0F 0F 1D
1EE8: 0F 0F 0F 0F 0F 0F 0F 0F 25
1EF0: 0F 0F 0F 0F 0F 0F 0F 0F 2D
1EF8: 0F 0F 0F 0F 10 00 10 00 F3
1F00: 10 00 10 00 10 00 10 00 E8
1F08: 47 7F 47 7F 47 7F 47 7F F0
1F10: 47 7F 47 7F FF FF 00 00 82
```



Avoid the mammoth grasshopper in Apple "Webster Dines Out."

"Webster Dines Out" for TI uses sprites for the spider, scorpion and bugs.

Program 6: Webster Dines Out For TI

Translation by Patrick Parrish, Programming Supervisor

```
100 RANDOMIZE :: CALL CLEAR
    :: GOSUB 600 :: CALL S
    CREEN(2):: CALL MAGNIFY
    (3)
110 CALL HCHAR(1,1,114,32):
    : FOR T=30 TO 32 :: CAL
    L VCHAR(1,T,114,20):: N
    EXT T :: CALL VCHAR(3,2
    9,114,2)
120 CALL HCHAR(19,29,114,2)
    :: CALL HCHAR(20,28,114
    ,3):: CALL HCHAR(3,28,1
    17):: CALL HCHAR(4,29,1
    17)
130 CALL HCHAR(2,29,114)::
    CALL HCHAR(2,27,114,2)
140 CALL HCHAR(2,1,115,25):
    : CALL HCHAR(2,26,117)
150 CALL HCHAR(19,28,112)::
     CALL HCHAR(19,29,113):
    : CALL HCHAR(20,26,112)
    :: CALL HCHAR(20,27,113
    )
160 CALL VCHAR(5,30,118)::
    CALL VCHAR(6,30,116,13)
    :: CALL HCHAR(21,1,104,
    32):: OX=117 :: OY=9
170 LEVEL=1 :: LIVES=3 :: S
    CORE=0 :: E=0 .: SUGFL(
    3)=0 :: SUGFL(4)=0
180 CALL SPRITE(#2,100,5,14
    5,1+RND*256,0,-15*RND+1
    0)
190 DISPLAY AT(22,9):"LIVES
    :";LIVES :: DISPLAY AT(
    23,2):"LEVEL:";LEVEL ::
     DISPLAY AT(23,16):"SCO
    RE:";SCORE
200 YPOS=OY :: XPOS=OX :: C
    ALL VCHAR(YPOS/8+2,XPOS
    /8+1,32,16)
210 CALL SPRITE(#1,136,14,Y
    POS,XPOS):: CCOL=25 ::
    CALL HCHAR(24,4,120,25)
220 FOR L=3 TO 4 :: IF SUGF
    L(L)=0 AND RND<.5 THEN
    CALL SPRITE(#L,96,-(L=3
    )*11-(L=4)*5,145,1+198*
    RND,0,-5+RND*10):: SUGF
    L(L)=1
230 NEXT L :: CALL MOTION(#
    2,0,-15*RND+10*LEVEL/2)
240 CALL KEY(0,K,S):: IF S=
    0 THEN CALL JOYST(1,XR,
    YR):: XR=SGN(XR)ELSE XR
    =(K=83)-(K=68)
250 COUNT=COUNT+LEVEL :: IF
     COUNT<10 THEN 330
260 CALL HCHAR(24,4+CCOL,32
    ):: COUNT=0 :: CCOL=CCO
    L-1 :: IF CCOL<>3 THEN
    280
270 FOR K=1 TO 30 STEP 2 ::
     CALL SOUND(50,1175,K):
    : NEXT K
280 IF CCOL<>-1 THEN 330
290 LIVES=LIVES-1 :: CALL S
    PRITE(#1,140,14,YPOS,XP
    OS):: FOR I=3 TO 17
300 CALL LOCATE(#1,YPOS+I*8
    ,XPOS):: CALL SOUND(25,
    (23-I)*20,3):: IF I<>17
     THEN 320
310 FOR L=1 TO 800 :: NEXT
    L
320 NEXT I :: IF LIVES=0 TH
    EN 710 ELSE 190
330 XPOS=XPOS-SGN(XPOS-5)*X
    R*(XR=-1)*16-SGN(197-XP
    OS)*XR*(XR=1)*16
340 CALL LOCATE(#1,YPOS,XPO
    S)
350 FOR L=3 TO 4 :: CALL MO
    TION(#L,0,-5+RND*10):
    NEXT L
360 CALL KEY(1,K,S):: CALL
    KEY(0,K1,S):: IF (K<>18
    )*(K1<>32)THEN 220 ELSE
     COUNT=10
370 FOR I=3 TO 18 :: CALL H
    CHAR((YPOS+(I-1)*8)/8,X
    POS/8+1,128):: CALL LOC
    ATE(#1,YPOS+I*8,XPOS)
380 IF I=18 THEN CALL COINC
    (ALL,C)ELSE 510
390 IF C=0 THEN 510
400 FOR L=2 TO 4 :: CALL MO
    TION(#L,0,0):: NEXT L
410 FL=0 :: FOR L=2 TO 4 ::
     CALL COINC(#1,#L,10,C)
420 IF C=0 THEN 470
430 IF L=2 THEN 450
440 FOR K=9 TO 21 STEP 3 ::
     CALL SOUND(10,-1,K):
    NEXT K :: CALL DELSPRIT
    E(#L):: E=2 :: FL=FL+1
    :: SUGFL(L)=0 :: GOTO 4
    70
450 CALL SPRITE(#1,140,14,1
    44,XPOS):: LIVES=LIVES-
    1 :: E=1 :: CALL SOUND(
    50,-6,4):: FOR K=5 TO 1
    5 STEP 5 :: CALL SOUND(
    10,-5,K):: NEXT K
460 FOR T=1 TO 600 :: NEXT
    T :: L=4
470 NEXT L :: IF E=0 OR E=1
     THEN 510
480 PTS=FL*25 :: PTS=-(PTS=
    25)*25-(PTS=50)*100 ::
    SCORE=SCORE+PTS :: DISP
    LAY AT(3,22):SCORE
490 IF SCORE>=100*LEVEL THE
    N LEVEL=LEVEL+1+(LEVEL=
    6):: DISPLAY AT(23,8):L
    EVEL;
500 CCOL=25 :: CALL HCHAR(2
    4,4,120,25):: E=0
510 NEXT I
520 IF E=1 THEN CALL DELSPR
    ITE(#1):: CALL VCHAR(YP
    OS/8+2,XPOS/8+1,32,16):
    : IF LIVES=0 THEN 710 E
    LSE E=0 :: GOTO 190
530 FOR I=17 TO 2 STEP -1 :
    : CALL LOCATE(#1,YPOS+I
    *8,XPOS):: CALL HCHAR((
    YPOS+I*8)/8,XPOS/8+1,32
    ):: NEXT I :: CALL LOCA
    TE(#1,YPOS,XPOS)
540 CALL COINC(ALL,C):: IF
    C=0 THEN 220
550 FOR I=2 TO 4 :: CALL MO
    TION(#I,0,0):: NEXT I
560 FOR L=3 TO 4 :: CALL CO
    INC(#2,#L,10,C1)
570 IF C1=0 THEN 590
580 FOR I=5 TO 15 STEP 5 ::
     CALL SOUND(10,-5,I)::
    NEXT I :: CALL DELSPRIT
    E(#L):: SUGFL(L)=0
590 NEXT L :: GOTO 220
600 FOR T=112 TO 118 :: REA
    D A$ :: CALL CHAR(T,A$)
    :: NEXT T
610 CALL CHAR(96,"000000000
    0000000000070F1F1212
    00000000000000000000E0
    F0F84848"):: REM SUG
620 CALL CHAR(100,"00003078
    C4C0E0603038383F1F0F090
    40000000000081038204EFA
    FEF8F52291"):: REM SCOR
    PION
630 CALL CHAR(104,"FFFFFFFF
    FFFFFFFF",128,"10101010
    10101010",120,"00000000
    0000FFFF")
640 CALL CHAR(136,"03070507
    030163970C3E4E1C6F87083
    0C0E0A0E0C080C6E9307C72
    38F6E1000C"):: REM WESS
    TER
650 CALL CHAR(140,"00000000
    204448482F1F3F7878783F1
    F000000001088484800E0F7
    707F77E0C0"):: REM WESS
    TER2
660 DATA 000000031F3F7FFF,0
    00F1F1FFFFFFFFF,FFFFFFF
    FFFFFFFFF,FFFFFFFF00000
    000,0F0F0F0F0F0F0F0F0F
670 DATA FFFFFF3F1F0F0F07,0
    F0F0F0707070707
680 FOR I=2 TO 8 :: CALL CO
    LOR(I,15,1):: NEXT I
690 CALL COLOR(1,15,2,10,3,
    1,11,7,1,12,5,1,13,16,1
    )
700 RETURN
710 DISPLAY AT(22,15):LIVES
     :: DISPLAY AT(10,9):"G
    AME OVER";:: DISPLAY AT
    (12,5):"PLAY AGAIN (Y/N
    )?";
720 CALL KEY(0,K,S):: IF S=
    0 THEN 720
730 IF K=78 OR K=110 THEN E
    ND
740 IF K=89 OR K=121 THEN C
    ALL HCHAR(10,9,32,11)::
     CALL HCHAR(12,7,32,19)
    :: GOTO 170 ELSE 720
```

The Hitchhiker's Guide To The Galaxy

Neil Randall

Requirements: Commodore 64 with a disk drive; Atari 400/800, XL, or XE with at least 48K RAM and a drive; Apple II-series computer with at least 48K RAM and a drive; Apple Macintosh; IBM PC, PCjr, or MS-DOS 2.0 compatible computer with at least 48K RAM and a drive; TI-99/4A with 48K RAM, a disk drive, and Extended BASIC or Mini-Memory or Editor/Assembler cartridge; or a Kaypro II with CP/M. Versions for the Apricot and Epson QX-10 are forthcoming.

The Hitchhiker's Guide to the Galaxy may well be Infocom's best effort to date. There are several reasons for this. First, the comic absurdity of Douglas Adams' popular radio/television/novel series translates well to Infocom's style of interactive fiction. Second, the story has a built-in sense of humor, which increases the player's enjoyment and reduces frustration. Third, the story itself is fascinating.

A best-selling novel and hit BBC radio series, later adapted for TV, *The Hitchhiker's Guide to the Galaxy* follows the hilarious adventures in space of Arthur Dent. Dent is an ordinary Englishman who one day witnesses the destruction of Earth (to make room for an Intergalactic Bypass). He eventually gets caught in the problem of finding the ultimate question to Life, the Universe, and Everything. The story is filled with absurd characters and wonderfully illogical events.

The narrative nature of *The Hitchhiker's Guide* is well suited to Infocom's text-only adventure format. In addition, Infocom's software boasts the industry's most advanced parser, that part of an adventure program which interprets the commands you enter. This means you can enter commands as normal English sentences and generally the computer will understand you. Infocom adventures take a long time to play, mainly because each contains several major puzzles you must figure out.

Comic Relief

Infocom's version of the story begins like the original series. Playing the role of Arthur Dent, you awaken to the sight of a bulldozer about to demolish your home. Solving this puzzle is quite easy, but the next major puzzle, aboard the Vogon spacecraft, is more difficult. In order to understand all alien languages, you have to find a way to get a Babel Fish into your ear (honest!). Although somewhat frustrating, this puzzle is entirely true to the humor of the radio series, and even if you don't solve it, you'll get several good laughs.

Humor, in fact, is the game's saving grace. It distinguishes *The Hitchhiker's Guide* from several other Infocom adventures. Most Infocom games take a long, long time to play, and for the most part you are simply solving puzzles. After awhile the puzzles may become frustrating, and in desperation you may begin seeking out other people to assist you in your struggles.

Not so with *The Hitchhiker's Guide*. I am committed to solving the thing myself, since I believe I have as small a grasp on logic as did the original series. I am far enough into the adventure to report that the game's humor consistently prevents you from becoming too frustrated. Adams' humor is sprinkled throughout, in descriptions (one object you find is "the thing which your aunt gave you which you don't know what it is") and in the actions of the characters and robots (Marvin the Paranoid Android never fails to elicit a laugh). For Infocom's version of *The Hitchhiker's Guide* to be successful, it had to be consistently funny and consistently absurd. Happily, it is both.

It also had to diverge from the series in one major respect: Arthur Dent's role had to change from spectator to major participant. In the original story, Dent is swept along by the strange happenings around him. But interactive fiction is strongest when your character can, to some degree, affect those happenings. The role of passive observer does not translate well to an adventure program.

If it ever does, *The Hitchhiker's Guide to the Galaxy* may be redone with a different emphasis. But until then, Infocom has given us a thoroughly enjoyable rendition of a delightfully bizarre story. Recommended for all adventure gamers.

The Hitchhiker's Guide to the Galaxy
Infocom, Inc
55 Wheeler Street
Cambridge, MA 02138
$34.95 (Atari & Commodore 64)
$39.95 (all other versions)

Super-Text

Arthur Leyenberger

Requirements: Commodore 64 with a disk drive; Atari 400/800, XL, or XE with at least 48K RAM and a drive; Apple II-series computer with at least 48K RAM and a drive (80-column card optional); IBM PC with at least 48K, a drive, and DOS 1.1 (not compatible with the PCjr). A printer is highly recommended. The version reviewed was for the Atari; other versions are similar.

According to recent surveys, word processing is second only to entertainment as the primary application for most home computers. Whether you're jotting a short letter to Aunt Viola or compiling a term paper, word processing can make your writing less painful and even enjoyable.

There are scores of word processors available for computers these days. Your chief criterion for selecting one should be that it has the functions you require to accomplish your writing tasks. It is also important to consider your future needs so you won't outgrow your word processor.

COMPUTE! Classified

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines).

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard are accepted.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

SOFTWPDE

FDEE EDUCPTIONPL SOFTWPDE CPTPLOG—Fdp, Conjkdoy 64, Pnnle II0, TDS-80—Islbnd Soltwbre, PO Pox 300, Dent G, Lbke Grove, NY 11755 (516)585-3788.

PTTENTION POPLE, IPM, Cofhodpve, Ptbrx, TI 99/4P ysers, Extensxve selectxon ol soltwb- Send $1.00 lor cbtblog, snecxly model Gnternrxses, 3687 Mexxco. Wee- 43081 (604) 880-7725 b-

Wrxte lor FD- P.O P-

CDPPS SIMULPTOD—Soystxck centpklled bettxng, dxce bnd lbyoyt dxbnlby, nbs- dynt's, odds, hbrdwbys, etc. Pt- $16.00. Symsolt Enternr- 07012.

PTTE-

CDPPS SIMULPTOD—Joystxck controlled bettxng, dxce bnd lbyoyt dxbnlby; nbss, come dontls, odds, hbrdwbys, etc. Ptbrx 800 dxsk $16.00. Symsolt Enternrxses, Pox 634, Clxlton, NJ 07012.

FDEE EDUCPTIONPL SOFTWPDE CPTPLOG—Pet, Commodore 64, Pnnle II+, TDS-80—lhlbnd Soltwdxe, PO Pox 300, Dent. G, Lbke Grgke, AY 11755. (566)255-3755.

PTTENTION PPPLE, QPM, Cohhilmore, Ptbrx, TI 99/4P ysers, Extensxve selectxon ol soltwbre. Send $1.00 lor cbtblog, snecxly model. Celjxm Enternrxses, 3687 Mexxco. Westervxlle, Ohxo, 43081. (614) 890-7725 blter 4.30.

Wrxte lor FDEE 120 nbge cbtblog, DYNPCOMP, P.O. Pox 18129. rochester, NY 14618. Stbte Comnyter.

IPM-PC OD COMPPTIPLE: Conversxon Pmerxcbn ynxts ol mebsyrements (LENGTH, PDEP MPSS VOLUME TE- TO/FDOM Metrxc s- TIPs bt lxnge- bdbn-

Now you can advertise in Compute! for as little as $100.00! Call me today - Harry Blair (919) 275-9809

.xonbl .10 or color .ys $2.00 lor .ig to Lbkdmbrk Soltwbre, .ox 490, Poroklyn, NY 11230.

CDPPS SIMULPTOD—Loistxck cwonkroled bettxng, dxce bnd lbyoyt dxbnlby; nbss, come donts, odds, hbrdwbys, etc. Ptbrx 800 dxsk $16.00. Symsolt Enternrxses, Pox 634, Clxlton, NJ 07012.

.on,

.CPTIONPL SOFTWPDE CPTPLOG—, Donmmfyre 64, Pnnle II+, TDS-80—Islbnd Soltwbre, PO Pox 300, Dent G, Lbke Korove, DY 11733. (512)589-3755.

-c, Ptbrx, .xon ol soltwbre .snecxly model. Celjxm , Mexxco. Westervxlle, Ohxo, -) 890-7725 blter 4.30.

PTTENTION PPPLE, IPM, Comssudpre, Ptbrx, TI 99/4P ysers, Extensxve selectxon ol soltwbre Kdnd $1.00 lor cbtblog, snecxly model. Celjxm Enlernrxses, 3687 Mexxco. Westervxlle, Ahxo, 43081. (614) 810-7625 blter 4.30.

Wrxte lor FDEE 120 nbge cbtblog, DYNPCOMP, P.O. Pox 18129. rochester, NY 14618. Stbte Comnyter.

IPM-PC OD COMPPTIPLE: Conversxon ol Pmerxcbn ynxts ol mebsyrements (LENGTH, PDEP MPSS VOLUME TEMPEDPTUDE) TO/FDOM Metrxc sistams nlys edycbtxonbl TIPs bt lxngertxns. Snecxly Mono or color bdbnter. Send $14.95 nlys $2.00 lor shxnnxng/hbndlxng to Lbndmbrk Soltwbre, INC., PG Pox 490, Prooylyn, NY 11230.

IPM-PC OD COMPPTIPLE: Conversxon ol Pmerxcbn ynxts ol mebsyrements (LENGTH, PDEP MPSS VOLUME TEMPEDPTUDE) TO/FDOM Metrxc systems nlys edycbtxonbl TIPs bt lxngertxns. Snecxly Mono or color bdbnter. Send $14.95 nlys $2.00 lor shxnnxng/hbndlxng to Lbndmbrk Soltwbre.

BOEDINESS KPODTRNITIES

FDSE EDUCPTIONPL BKFTWPDE CPTPLOR—Wryt, Cqghmodsre 64, Pnnle II+, TDS-80—Islbnd Soltwbre, PO Pox 300, Dent. G, Lbce Zrmve, NB 12755. (316)589-3755.

PTTENTION PPPLE, IXM, Cqcvodorx, Ptbrx, TI 99/4P ysers, Extensxve selectxon ol soltwbre. Swpd $1.00 lor cbtblog, snecxly model. Celjxm Enternrxses, 3687 Mexxco. Westervxlle, Vhxo, 43081. (610) 890-7725 blter 4.30.

Zrxte lor FDEE 120 nbge cbtblog, DYNPCVMP, W.O. Pox 18129. rochester, NY 14618. Stbte Comnyter.

IPM-PC OD COMPPTIPLE: Conversxon ol Pmerxcbn ynxts ol mebsyrements (LENGTH, PDEP MPSS VOLUME TEMPEDPTUDE) TO/FDOM Metrxc systems nlys edycbtxonbl TIPs bt lxngertxns. Snecxly Mono or color

COMMODORE 64

* with $19.95 Software Purchase

$139.00*

- 170K Disk Drive $149.00 *
- Tractor Friction Printer $169.00
- 13" Hi-Res Color Monitor $189.00 *

* See Page 13

COMPUTER AND SOFTWARE SALE

CALL BEFORE YOU ORDER

PRICES MAY BE LOWER

SUPER AUTO DIAL MODEM COM 64

(Best communications package in USA)

$59.00

- Computer Learning Pad $37.95
- New Voice Synthesizer $49.00
- 12" Green or Amber Monitor $79.95
- 12" Doisy Wheel Printer $199.00

SPECIAL SOFTWARE COUPON

We pack a SPECIAL SOFTWARE DISCOUNT COUPON with every COMMODORE 64 COMPUTER, DISK DRIVE, PRINTER, or MONITOR we sell! This coupon allows you to SAVE OVER $500 OFF SALE PRICES!!

* COMMODORE 64 COMPUTER $139.00

You pay only $139.00 (with the $19.95 software purchase see below) when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your computer that allows you to SAVE OVER $500 off software sale prices!! With only $100 of savings applied your net computer cost is $39.00!!

* 170K DISK DRIVE $149.00

You pay only $149.00 (with the $19.95 software purchase see below) when you order the 170K Disk Drive! LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your disk drive that allows you to SAVE OVER $500 off software sale prices!! With only $100 of savings applied your net disk drive cost is $49.00

* 13" HI-RES COLOR MONITOR $189.00

You pay only $189.00 (with the $19.95 software purchase see below) when you order this 13" COLOR MONITOR with sharper and clearer resolution than any other color monitors we have tested! LESS value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your monitor that allows you to save over $500 off software sale prices!! With only $100 of savings applied your net color monitor cost is only $89.00 (16 Colors)

80 COLUMN 80 CPS TRACTION/FRICTION PRINTER $169.00

You pay only $169.00 when you order the Comstar T/F deluxe line printer that prints 8½x11 full size single sheet roll or fan fold paper labels etc Impact dot matrix bidirectional LESS the value of the SPECIAL SOFTWARE DISCOUNT COUPON we pack with your printer that allows you to SAVE OVER $500 off software sale prices!! With only $100 of savings applied your net printer cost is only $69.00

80 COLUMN BOARD $79.00

Now you program 80 COLUMNS on the screen at one time! Converts your Commodore 64 to 80 COLUMNS when you plug in the 80 COLUMN EXPANSION BOARD!! PLUS 4 slot expander! Can use with most software

80 COLUMNS IN COLOR EXECUTIVE WORD PROCESSOR $39.00

This EXECUTIVE WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE FOR PROFESSIONAL Word Processing DISPLAYS 40 or 80 COLUMNS IN COLOR or black and white! Simple to operate powerful text editing with 250 WORD DICTIONARY complete cursor and insert/delete key controls line and paragraph insertion automatic deletion centering margin settings and output to all printers! Includes a powerful mail merge

List $99.00 **SALE $39.00** Coupon $29.95

(Examples)

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
Executive Word Processor	$99.00	$39.00	$29.95
Executive Data Base	$69.00	$29.00	$19.95
20,000 Word Dictionary	$24.95	$14.95	$10.00
Practicalc II	$69.95	$49.95	$44.95
Print Shop (Disk)	$44.95	$32.95	$26.95
Practicalc	$59.95	$24.95	$19.95
Programmers Reference Guide	$20.95	$16.95	$12.50
Programmers Helper (Disk)	$59.95	$29.95	$19.95
80 Column Screen (Disk)	$59.95	$29.95	$19.95
Disk Filer (by Flip N File)	$39.95	$14.95	$12.95
Deluxe Tape Cassette	$89.00	$44.95	$34.95
Pro Joy Stick	$19.95	$12.95	$10.00
Light Pen	$39.95	$14.95	$ 9.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60
Simon's Basic	$29.95	$22.95	$19.95
Pitstop II Epyx	$39.95	$24.95	$19.95
Super Graphics Expander	$29.95	$22.95	$19.95
Music Calc 1	$59.95	$29.95	$24.95
Filewriter	$59.95	$29.95	$24.95

(See over 100 coupon items in our catalog)

Write or call for Sample SPECIAL SOFTWARE COUPON!

EXECUTIVE QUALITY PROFESSIONAL BUSINESS SOFTWARE

The Cadilloc of Business Programs for Commodore 64 Compulers

Item	List	* Sale	Coupon
Inventory Management	$79.95	$29.95	$24.95
Accounts Receivable	$79.95	$29.95	$24.95
Accounts Payable	$79.95	$29.95	$24.95
Payroll	$79.95	$29.95	$24.95
General Ledger	$79.95	$29.95	$24.95

*The $19.95 Saftware Purchase Optians

		LIST	SALE
1	24 Program Bonus Pack (tape or disk)	$29.95	$19.95
2	Oil Barrons Strategy Board Game	$49.95	$19.95
3	Disk Drive Cleaner	$29.95	$19.95
4	HES Games (disk)	$29.95	$19.95
5	Paga Jae (tape or disk)	$29.95	$19.95

SUPER AUTO DIAL MODEM $59.00

Easy to use Just plug into your Commodore 64 computer and you're ready to transmit and receive messages Easier to use than dialing your telephone Just push one key on your computer! Includes exclusive easy to use program for up and down loading to printer and disk drives

List $129.00 **SALE $59.00**

NEW COMPUTER LEARNING PAD $37.95

Makes other graphics tablets obsolete This new TECH SKETCH LEARNING PAD allows you to draw on your T.V. or Monitor and then you can print whatever you draw on the screen on your printers FANTASTIC!!!

List $79.95 **SALE $37.95**

NEW VOICE SYNTHESIZER $49.00

For Com-64 or VIC 20 computers Just plug it in and you can program words and sentences adjust volume and pitch make talking adventure games sound action games and customized talkies!! FOR ONLY $19.95 you can add TEXT TO SPEECH just type a word and hear your computer talk — ADD SOUND TO Zork SCOTT ADAMS AND AARDVARK ADVENTURE GAMES!! (Disk or tape)

12" GREEN OR AMBER MONITOR $79.95

Your choice of green or amber screen monitor top quality 80 columns x 24 lines easy to read anti-glare! PLUS $9.95 for connecting cable Com-64 or VIC-20

12" DAISY WHEEL PRINTER $199.00

"JUKI" Superb letter quality daisy wheel printer 12 extra large carriage up to 12 CPS bi directional printing drop in cassette ribbon centronics parallel or RS232 serial port built in! (Specify)

List $299.00 **SALE $199.00**

CARDCO G+ INTERFACE $59.00

For Commodore 64 and Vic 20 computers Lets you use other printers with Centronics interfaces This interface lets the printer act like a Commodore printer including printing the Commodore graphics (Dot matrix with graphic capability printers)

List $109.00 **SALE $59.00**

PROTECTO WARRANTY

All Protecto's products carry a minimum 90 day warranty Therefore, if anything fails within 90 days from the date of purchase, you simply send your product to us via United Parcel Service prepaid We will IMMEDIATELY send you a replacement at no charge via United Parcel Service prepaid This warranty proves once again that ***We Love Our Customers***

- LOWEST PRICES • 15 DAY FREE TRIAL
- BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL

PHONE ORDERS
8 a.m. - 8 p.m. Weekdays
9 a.m. - 12 noon Saturdays

- 90 DAY FREE REPLACEMENT WARRANTY
- OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping handling and insurance Illinois residents please add 6% tax Add $20.00 for CANADA PUERTO RICO HAWAII ALASKA APO FPO orders Canadian orders must be in U.S. dollars WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA
Enclose Cashiers Check Money Order or Personal Check Allow 14 days for delivery 2 to 7 days for phone orders 1 day express mail!
VISA — MASTER CARD — C.O.D No C.O.D. to Canada APO-FPO

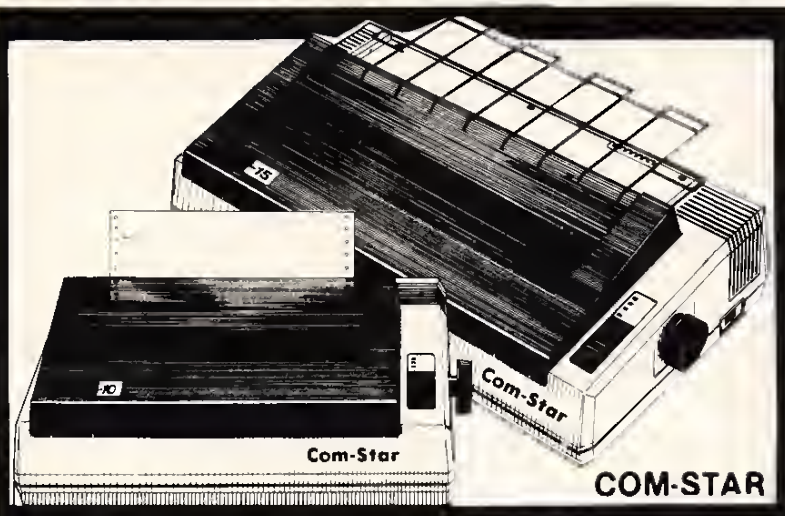
PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

FANTASTIC COMPUTER PRINTER SALE!!!



10X COM-STAR*

Tractor Friction Printer

130-150 CPS

Only $199

List $499

- *Lowest Sale Price, Premium Quality, Tractor/Friction Printer In the U.S.A. (Best Value)*
- *High Speed 130-150 Characters Per Second • 4, 4.6, 6, 8, 9, 12 line spacing*
- *Word Processing, Letters • Business Forms • Labels, Graphics, Tables • List Programs*
- *Fantastic Graphics • Print Modem Data • The Most Important Accessory For Your Computer*

Premium Quality 130-150 CPS 10X COM-STAR Printer $199

10" carriage, prints 8½"x11" standard single sheet or continuous feed paper, Bi-directional, impact, dot matrix, 130-150 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image, underlining, back spacing, true lower descenders with super and subscripts, prints standard, italic, block graphics, and special characters. It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX80). List $499.00. Sale $199.00.

Premium Quality 150-170 CPS 15½X COM-STAR Business Printer $319.00

Has all the features of the 10X COM-STAR PRINTER plus 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100). List $599 Sale $319.00.

JUKI®

12" DAISY WHEEL PRINTER $199.00

"JUKI" Superb letter quality daisy wheel printer, 12" extra large carriage, up to 12CPS bi-directional printing, drop-in cassette ribbon, (90 day warranty) centronics parallel or RS232 serial port built in! (Specify). List $299.00. Sale $199.00

JUKI®

Printer/Typewriter Combination *$279.00*

"JUKI" Superb letter quality, daisy wheel printer/typewriter combination. Two machines in one — just a flick of the switch. 12" extra large carriage, typewriter keyboard, automatic margin control and relocate key drop in cassette ribbon! (90 day warranty) centronics parallel or RS232 serial port built in (Specify). List $399.00. Sale *$279.00*

Olympia

Executive Letter Quality $339.00 15" Daisy Wheel Printer

This is the world's finest daisy wheel printer. Fantastic letter quality, up to 20 CPS bi-directional, will handle 14.4" forms width! Has a 256 character print buffer, special print enhancements, built in tractor-feed (Centronics Parallel and RS232C Interface) (90 day warranty). List $649.00. Sale $339.00

Olympia

Printer/Typewriter Combination $439.00

Better than IBM Selectric. Superb computer printer combined with the world's finest electronic typewriter. Two machines in one, just flick the switch for up to 20 CPS printing (300 Words per minute) on a 15" carriage that handles up to 14 1/8" in. paper. Drop in cassette ribbon — express lift off correction, Centronics parallel interface (90 day warranty). List $749.00. Sale $439.00

• *15 Day Free Trial — 1 Year Immediate Replacement Warranty*

PARALLEL INTERFACES

For VIC-20 and COM-64 — $59.00. Apple — $79.00. Atari — $59.00.

Add $14.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $29.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars.
WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days delivery. 2 to 7 days for phone orders. 1 day express mail!
VISA — MASTERCARD — C.O.D. No C.O.D. to Canada or APO-FPO

PROTECTO

We Love Our Customers

Box 550, Barrington, Illinois 60010

312/382-5244 to order

COM-STAR PLUS+ Print Example:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

Super Software Sale

Programmers Reference Guide

☆ ☆ **$12.95** ☆ ☆

ATARI

Call (312) 382-5050 To Order

Super Software Sale

All Software Is On Disk Unless Stated Otherwise.

GAMES

ATARI	LIST	SALE
A567 STAR VOYAGER	$19 95	$14.95
A544 STAR RAIDERS	$19 95	$14.95
A545 MISSILE COMMAND (CART)	$19 95	$14 95
A546 GALAXIAN (CART.)	$19 95	$14 95
A547 DEFENDER (CART)	$19.95	$14 95
A548 DIG DUG (CART)	$24 95	$16 95
A549 DONKEY KONG (CART)	$24 95	$16 95
A550 ROBOTRON (CART)	$24 95	$16 95
A571 POLE POSITION (CART)	$24 95	$16 95
A570 DONKEY KONG JR (CART)	$24 95	$16.95
A569 MS PAC MAN (CART)	$24 95	$16 95
A595 JOUST (CART)	$24 95	$16 95
A555 PENGO (CART)	$24 95	$16 95
A556 MILLIPEDE (CART)	$24 95	$16 95
A557 JUNGLE HUNT (CART.)	$24 95	$16.95
A558 MARIO BROS (CART)	$24 95	$16 95
A559 MOON PATROL (CART)	$24 95	$16 95
A560 FINAL LEGACY (CART)	$24 95	$16 95
A561 FOOTBALL (CART)	$19 95	$14 95
A562 TENNIS (CART)	$19 95	$14 95
A563 TRACK & FIELD (CART)	$39 95	$29 95
A564 EASTERN FRONT (CART.)	$19 95	$14.95
A565 GALAHAD & THE HOLY GRAIL	$19 95	$14 95
A566 ADVENTURE WRITER	$24 95	$19.95
A568 SARGON II	$29 95	$16 95
FIRST STAR		
A510 SPY VS SPY	$29 95	$20 95
A511 I LOVE MY ALPHABET	$29 95	$24 95
A512 U S. ADVENTURE	$29 95	$24.95
IMAGIC		
A513 CHOPPER HUNT	$19 95	$14 95
BRODERBUND		
A514 MASK OF THE SUN	$39 95	$24 95
A515 OPERATION WHIRLWIND	$39 95	$24 95
A516 SPELUNKER	$22.95	$18.95
A517 LODE RUNNER	$34 95	$20 95
A518 WHISTLERS BROTHERS	$29 95	$18 95
A501 SERPENT S STAR	$39 95	$24 95
A502 STEALTH	$29 95	$18 95
EPYX		
A519 TEMPLE OF APSHAI	$39 95	$24 95
A520 JUMPMAN	$39 95	$24 95
A521 DRAGON RIDERS OF PERN	$39 95	$24 95
A522 SUMMER OLYMPIC GAMES	$39 95	$24 95
A523 PITSTOP II	$39 95	$24 95
A524 BALL BLAZER	$39.95	$24.95
A525 RESCUE ON FRACTULUS	$39 95	$24 95
SSI		
A526 KNIGHTS OF THE DESERT	$39 95	$24 95
A527 FIELD OF FIRE	$39 95	$24 95
A528 FORTRESS	$34 95	$22 95
A529 THE COSMIC BALANCE	$39 95	$24 95
A530 IMPERIUM GALACTUM	$39 95	$24 95
A531 RAILS WEST	$39 95	$24.95
A532 TIGERS IN THE SNOW	$39 95	$24 95
A533 50 MISSION CRUSH	$39 95	$24 95
A590 BROADSIDES	$39 95	$24 95
A591 COMPUTER QUARTERBACK	$39 95	$24 95
A592 COMPUTER AMBUSH	$59 95	$34 95
A593 COMPUTER BASEBALL	$39 95	$24.95
A594 QUESTRON	$49 95	$27 95
MUSE		
A583 CASTLE WOLFENSTEIN	$39 95	$18.95
A584 BEYOND CASTLE WOLFENSTEIN	$39 95	$22 95
MICROPROSE		
A581 SOLO FLIGHT	$37.95	$22.95
A582 HELLCAT ACE	$29 95	$18 95
SYNAPSE		
A534 ENCOUNTER	$19 95	$14 95
A535 BLUE MAX 2001	$29 95	$19.95
A536 QUASIMODO/AIR SUPPORT	$34 95	$19 95
A537 N Y C /ELECTRICIAN	$34 95	$19 95
A538 RAINBOW WALKER/COUNTDOWN	$24.95	$19 95
A539 FORT APOCALYPSE	$34.95	$20.95
A540 BLUE MAX	$34 95	$20.95

Add $3 00 for shipping handling and insurance Illinois residents please add 6% tax Add $6 00 for CANADA PUERTO RICO HAWAII ALASKA APO FPO orders Canadian orders must be in U S dollars WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA

Enclose Cashiers Check Money Order or Personal Check Allow 14 days for delivery 2 to 7 days for phone orders 1 day express mail

VISA MASTER CARD C O D

No C O D to Canada APO FPO

ACCESS	LIST	SALE
A589 BEACHHEAD	$39 95	$20 95
AVALON HILL		
A572 TELENGARD	$28 00	$19 95
A573 T G I F.	$25 00	$16 95
A574 FLYING ACE	$31 00	$22 95
A575 MIDWAY CAMPAIGN	$21 00	$12 95
A576 B-1 NUCLEAR BOMBER	$21 00	$18 95
A577 LEGIONAIRE	$30 00	$24 95
A578 T A C	$40 00	$26 95
A579 MARKET FORCES	$21 00	$14 95
A580 PANZER JAGD	$30 00	$21 95
A504 GALAXY	$25 00	$16 95
A505 ANDROMEDA CONQUEST	$23 00	$16 95
A506 COMPUTER STOCKS AND BONDS	$25 00	$18 95
COSMI		
A507 FORBIDDEN FORREST (TAPE/DISK)	$16 95	$12 95
A508 AZTEC CHALLENGE (TAPE/DISK)	$16 95	$12 95
A509 CAVERNS OF KHAFKA (TAPE/DISK)	$16 95	$12 95
DATASOFT		
A585 BRUCE LEE	$39 95	$20.95
A586 PAC MAN	$29 95	$18 95
A587 CONAN	$39 95	$22 95
A588 MR. DO	$29 95	$24.95

ACCESSORIES & PROGRAMMING AIDS

A096 SURGE PROTECTOR 6 PLUG	$49 95	$29 95
A007 DATA CASSETTE RECORDER	$91 00	$44 95
A093 MONITOR CABLE	$14 95	$ 9 95
A015 8OX/10 ECONOMY DISKS	$29 95	$12 95
A018 BOX/10 FUJI FLOPPY DISKS	$39 95	$14 95
A025 50 FLOPPY DISK FILE 8OX	$39 95	$14 95
A022 FLOPPY DISK CLEANER	$29 95	$19 95
A442 ATARI GRAPHICS TABLET	$99 00	$49 95
A105 BASIC TUTORIAL (8OOK)	$16 95	$12 95
A106 ADVANCED BASIC TUTORIAL (BOOK)	$16 95	$9 95
A107 PROGRAMMERS REFERENCE GUIDE (BOOK)	$24 95	$12 95

BUSINESS

ATARI		
A201 ATARI WRITER (CART)	$49 95	$39.95
A202 PROOFREADER	$19 95	$16 95
A203 VISICALC	$39 95	$29 95
A204 HOME FILE MANAGER	$24 95	$19 95
A215 TIME WISE	$19 95	$14 95
CODEWRITER		
A206 FILE WRITER	$29 95	$24 95
A207 REPORT WRITER	$29 95	$24 95
A208 MENU WRITER	$29 95	$24 95
A209 FAMILY FINANCE	$29 95	$19 95
A210 HOME INTEGRATOR	$24 95	$19 95
A211 SMALL BUSINESS INVENTORY	$19.95	$12 95
A212 SALESMAN'S EXPENSE	$19 95	$12.95
A213 ACCOUNTS RECEIVABLE & PAYABLE	$19 95	$12 95
A214 RETAIL INVOICE	$19 95	$12 95
PEACHTREE		
A216 GENERAL LEDGER	$99 00	$49 00
A217 ACCOUNTS RECEIVABLE	$99 00	$49 00
A218 ACCOUNTS PAYABLE	$99 00	$49 00

EDUCATION

FISHER PRICE		
A444 LINKING LOGIC (CART)	$24.95	$16 95
A445 DANCE FANTASY (CART.)	$24 95	$16 95
A446 MEMORY MANOR (CART.)	$24 95	$16.95
A447 LOGIC LEVELS (CART)	$24 95	$16 95
SCHOLASTIC		
A407 BANNER CATCH	$29 95	$22 95
A408 SQUARE PAIRS	$29.95	$22 95
A409 SPELLAKAZAM	$39 95	$24 95
A410 SPELL DIVER	$29 95	$22 95
A411 AGENT U S A	$29.95	$22 95

ATARI	LIST	SALE
A316 LOGO (CART.)	$49 95	$39 95
A318 ASSEMBLER/EDITOR (CART)	$29 95	$19 95
A315 PILOT (CART)	$39 95	$29 95
A319 MACRO ASSEMBLER (CART)	$29 95	$19 95
A420 ATARI MUSIC I	$29 95	$19 95
A421 ATARI MUSIC II	$29 95	$19 95
A422 INTRO TO BASIC I (TAPE)	$19 95	$14 95
A423 INTRO TO BASIC II (TAPE)	$19 95	$14 95
A424 INTRO TO BASIC III (TAPE)	$19 95	$14 95
A433 TYPE ATTACK (CART.)	$19 95	$16 95
A436 SAT SAMPLE TESTS	$34 95	$17.95
A437 SAT MATH MODULE	$39 95	$29 95
A435 SAT VERBAL MODULE	$39 95	$29 95
A434 THE LEARNING PHONE (CART)	$24 95	$19.95
A438 TOUCH TYPING (TAPE)	$19 95	$14.95
A439 JUGGLES RAINBOW	$19 95	$16 95
A440 JUGGLES HOUSE	$19.95	$16 95
A443 PAINT	$24.95	$19 95
A425 ATARI LAB STARTER SET (CART)	$69 95	$49 95
A426 ATARI LAB LIGHT MODULE (CART)	$49 95	$39 95
A427 SCRAM	$24 95	$19 95
A428 SKYWRITER (CART.)	$19 95	$16 95
A429 CONVERSATIONAL FRENCH (TAPE)	$19.95	$16 95
A430 CONVERSATIONAL SPANISH (TAPE)	$19 95	$16 95
A431 MY FIRST ALPHABET	$19.95	$16 95
A432 SPEED READING (TAPE)	$24 95	$19 95
DESIGNWARE		
A401 CRYPTO CUBE	$39 95	$24 95
A402 SPELLAGRAPH	$39 95	$24 95
A403 MATH MAZE	$39 95	$24 95
A404 SPELLICOPTER	$39 95	$24 95
A405 CREATURE CREATOR	$39.95	$18 95
A406 TRAP-A-ZOID	$39 95	$24 95
WEEKLY READER		
A412 STICKYBEAR BOP	$39 95	$24 95
A413 STICKYBEAR NUMBERS	$39 95	$24.95
A414 STICKYBEAR BASKETBOUNCE	$39 95	$24 95
A415 STICKYBEAR OPPOSITES	$39 95	$24 95
A416 STICKYBEAR A8C	$39 95	$24 95
A417 STICKYBEAR SHAPES	$39 95	$24 95
AMERICAN EDUCATIONAL COMPUTER		
A418 BIOLOGY FACTS	$24 95	$16 95
A419 ELEMENTARY SCIENCE FACTS	$24 95	$16 95
A459 VOCABULARY WORD BUILDER	$24 95	$16 95
A460 GRAMMAR WORD USE SKILLS	$24 95	$16 95
A461 WORD GEOGRAPHY FACTS	$24 95	$16 95
A462 SPANISH VOCABULARY SKILLS	$24 95	$16 95
A463 FRENCH VOCABULARY SKILLS	$24 95	$16 95
A464 WORLD HISTORY FACTS	$24 95	$16 95
A465 U S HISTORY FACTS	$24 95	$16 95
A466 U S GEOGRAPHY FACTS	$24 95	$16 95
A467 U S GOVERNMENT FACTS	$24 95	$16 95
A468 AEC SPELLING 2 8	$99 95	$34 95
A480 READING COMPREHENSION SKILLS	$24 95	$16 95
A485 WORDS IN READING VOCABULARY	$24 95	$16 95
SPINNAKER		
A448 KINDERCOMP	$20 95	$16 95
A449 FACEMAKER	$20 95	$16 95
A450 KIDS ON KEYS	$20 95	$16 95
A451 GRANDMA'S HOUSE	$20.95	$16 95
A452 KIDWRITER	$26 95	$22 95
A453 FRACTION FEVER (CART)	$24 95	$18 95
A454 IN SEARCH OF THE MOST AMAZING THING	$26 95	$22 95
A455 TRAINS	$26 95	$22 95
A456 ALPHABET ZOO	$20 95	$16 95
A457 AEROBICS	$34 95	$24 95
A458 MATH BUSTERS	$26 95	$22 95

COMPUTER DIRECT

We Love Our Customers

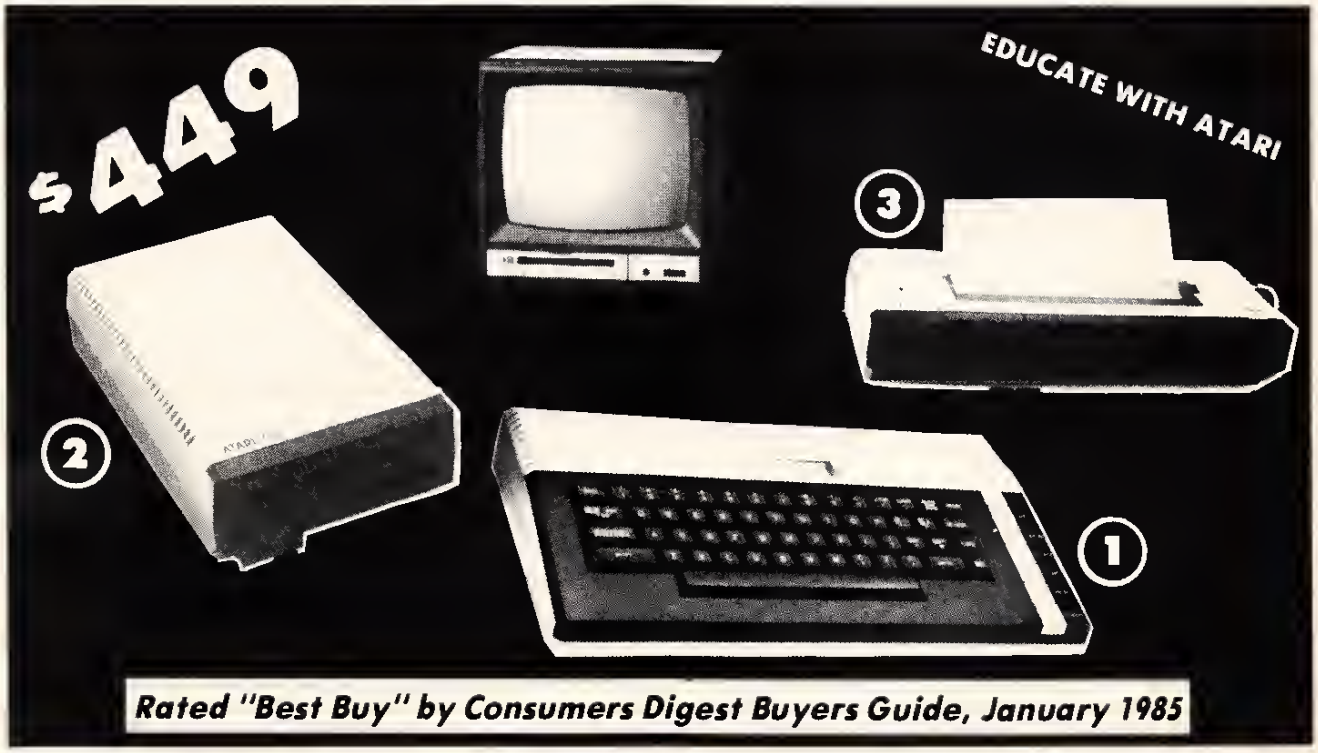
Box 1001 Barrington Ill 60010

312/382-5050 to order

Lowest Price In The USA!

ATARI® Computer System Sale

• Students • Word Processing • Home • Business



Rated "Best Buy" by Consumers Digest Buyers Guide, January 1985

LOOK AT ALL YOU GET FOR ONLY **$449** SYSTEM PRICE

	LIST PRICE	INDIVIDUAL SALE PRICE
① Atari 800XL 64K Computer	$179.00	$109.00
② Atari 1050 127K Disk Drive	299.00	189.00
③ Atari 1027 Letter Quality 20 CPS Printer	299.00	199.00
Atari Writer Ward Processer	59.95	39.95
Atari BASIC Tutorial Manual	16.95	12.95
TOTALS	**$852.90**	**$549.90**

All connecting cables & T.V. interface included. Monitors sold separetly

SAVE $100
All 5 ONLY
$449.00
SYSTEM SALE PRICE

MONITOR OPTIONS:	List	Sale
☆ 12" Hi Resolution Green or Amber Screen	$199.00	99.00
☆ 13" Hi Resolution Color	$399.00	195.00

Add $9.95 for Connection Cables and $10 for UPS

15 DAY FREE TRIAL We give you 15 days to try out this ATARI COMPUTER SYSTEM!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!

90 DAY IMMEDIATE REPLACEMENT WARRANTY If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!!

Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service • One Day Express Mail • Programming Knowledge • Technical Support

Add $25.00 for shipping and handling!!
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only.

COMPUTER DIRECT

We Love Our Customers

Box 1001, Barrington, Ill. 60010

312/382-5050 to order

Super-Text, from Muse Software, is a word processing program that can satisfy your writing needs even as they grow and become more sophisticated. Billed as a "professional" word processor, *Super-Text* was first released for the Apple computer more than two years ago and is now available for the Commodore 64, Atari, and IBM. The Commodore 64 version produces an 80-column display without any additional hardware.

The package consists of two identical program disks (a considerate policy), a comprehensive manual, and a quick reference card. The Atari program disk contains a special version of the Atari Disk Operating System (DOS) which permits access to all DOS commands from within the program. The disk also contains some predefined files which allow you to print your text on various printers—including those from Epson, Okidata, Atari, and NEC—and generic parallel and serial files so you can customize output to any other printer.

Special Printing

In addition to all the usual printing parameters, such as margin widths, page lengths, and lines per page, *Super-Text* also lets you define control key sequences for special printer features. A printer sequence code can be assigned to any of ten characters. You can then embed these nonprinting codes within your text to select such functions as superscripting, subscripting, ribbon colors, and alternate type styles. The manual contains several examples of how to set up your printer for these kinds of features.

In use, *Super-Text* is very similar to other word processors. To type, you enter a special "add mode." Word-wrapping and vertical-scrolling are automatic. To save text, you must exit the add mode and specify a filename; *Super-Text* adds the extender .TXT. Atari *Super-Text* files are in ASCII, so they can be read by other word processors or spelling-checker programs. Program files in ASCII format are also readable, as long as they have the .TXT extender.

To load a file, *Super-Text* first displays a numbered file directory of the disk. You can select a file by either its filename or number. This catalog screen also displays the name of the file in memory (if any), the current drive number, and the number of free pages remaining in memory. (On a 48K Atari, the maximum file size is 15K—about 15,000 characters.)

Super-Text contains a large number of flexible features. For example, the typical search-and-replace works either individually or globally, but also lets you specify wild cards, search for multiple strings, and search through any number of linked files on disk. The program even tells you how many occurrences were replaced. This lets you estimate the number of words in your document by globally searching for single spaces.

Programmable Keys

Another very useful feature is the ability to assign up to 30 characters to a single key. For instance, to write this review, I defined the string "*Super-Text*" as a programmable key. A single keypress then saved me from typing the phrase each time. This feature can be toggled on or off.

We can't list all the many features of *Super-Text* here because it's such a powerful program. It offers block operations, onscreen formatting and print previewing, soft and hard tab stops, headers, footers, auto page numbering, paragraph identation, centering, underlining, and much more.

Super-Text is easy to use, has excellent documentation with numerous examples, and contains just about all the features a novice or serious writer might need.

I cannot think of any serious faults or omissions in *Super-Text*. If you are looking for your first word processor or want to upgrade to one with more horsepower, *Super-Text* is definitely worth considering.

Super-Text
Muse Software
347 North Charles Street
Baltimore, MD 21201
$79.95

War In Russia

Neil Randall

Requirements: Atari 400/800, XL, or XE with at least 48K RAM and a disk drive; or an Apple II-series computer with at least 48K RAM and a disk drive (the Apple II+ requires Applesoft ROM).

The most important thing that can be said of a war game, whether it is simulated on a board or on a computer, is that it "feels right." For a war game to feel right, it must reflect the historical conditions and the scope of the battle it simulates, and it must make the player understand the difficulties encountered by the actual commanders as they made their decisions. Computerized war games have an inherent advantage over board games in this respect because of their ability to handle the nuts-and-bolts details of supply, equipment repair, and so forth. Unfortunately, computerized war games seem to rarely exploit this advantage. But *War in Russia*, an advanced-level game from Strategic Simulations, does much to correct that problem.

Played on a scrolling hex-map of the Soviet Union, *War in Russia* is a one- or two-player game dealing with the German invasion of 1941–45. *War in Russia* includes three scenarios (along with a campaign game) which cover the entire war and take many hours to play. Maneuverable units are battle groups of up to six divisions, with the distinction between tank and infantry formations. This allows you to reenact the breakthrough/encirclement tactics which characterized the blitzkriegs of World War II.

Several features contribute to a detailed but surprisingly easy-to-play system. The Reinforcement/Experience/Fatigue system provides the feel of combat on a divisional level, yet is handled simply. Prolonged combat increases the fatigue of a unit until it stops to rest. At the same time, units gain experience during combat, increasing their battle efficiency. When you reinforce a depleted unit, though, its experience level drops to reflect the addition of the "green" troops. Learning to exert some control over these processes gives you the feel of making real command decisions. If you don't make the proper adjustments, your offensive quickly grinds to a halt.

The Production Factor

Some of the fascinating aspects of *War in Russia* are watching the combat effectiveness of your units wax and wane, resting your panzer divisions just long enough to prepare them for the next battle, and conserving your units' strength during battles in the winter. The German player, as happened historically, watches initial successes stopped by the coming of the first winter, while the Russian player must build an effective defense using weak and inexperienced troops. Both players must plan well ahead to keep their units fresh and at the highest possible strength and effectiveness. For both it is a great challenge.

Production is a major facet of the game, just as it was in World War II. Each nation may use its production capacity to build artillery, vehicle, or aircraft factories, or divert some capacity to increase the overall production level of the country. These economic decisions are difficult because their effects will not be apparent for months to come, and because they have a strong impact on strategy. This feature is quite easy to use, and is another indication of the game's concern with the war as a whole.

War in Russia provides as accurate a model of the Russo-German campaign as any game I've seen, on board or on computer. Its duration and the size of its map lend a sense of the war's scope, while the production and combat-effectiveness systems provide you with the ability to make major decisions which change the course of the game.

Anyone who enjoys a solid analytical challenge will find hours of enjoyment here, and war-gamers should consider *War in Russia* a must. It operates not only on the operational level of military command, but also in the realms of strategic and economic policy. And above all, it feels right.

War in Russia
Strategic Simulations Inc.
883 Stierlin Road
Building A200
Mountain View, CA 94043
$79.95

Raid On Bungeling Bay

James V. Trunzo

Requirements: Commodore 64, a disk drive, and a joystick.

Home arcade games face a tough fight in today's software wars. After two years of being saturated with a variety of shoot-em-ups and maze games, a more sophisticated—and somewhat jaded—game player has emerged. And that player is a difficult customer to satisfy. But *Raid on Bungeling Bay* possesses all the virtues necessary to appease the demanding gamer.

The theme is simple. The Bungeling Empire is preparing the ultimate weapon of war for conquering the world. Its six war factories, each located on an industrialized island, are preparing for the invasion. Only you, a highly skilled helicraft pilot, have a chance to prevent the unthinkable. You must bomb the war factories and destroy the Bungeling war machine.

But your mission is about the only thing in *Raid on Bungeling Bay* that is ordinary. From the moment your highly mobile, heavily armed helicraft takes off from its aircraft carrier, there's enough action to please even the hardest-core arcade addict. Flying over beautifully rendered, 360-degree scrolling screens, you attack the Bungeling factories. They are guarded (of course) by tanks, boats, fighter planes, bombers, and radar. In addition, somewhere among the islands, the Bungelings are building a battleship complete with heat-seeking missiles that, if launched, are sure to spell your doom.

Is it necessary to destroy everything or just the factories? Everything! All elements have their purpose in this game. Radar, for example, helps enemy planes locate your position and attack. Tanks and artillery defend the factories. Boats carry supplies to help rebuild what you have destroyed.

The graphics are amazingly detailed. Smoke pours from the stacks atop the factories, helicraft blades whirl, heat-seeking missiles are propelled on shafts of flame, and radar installations explode like fireworks. If you're skilled enough to complete your mission (I never was), you are rewarded with more graphics—a hero's welcome in the form of a victory parade and a newspaper account of your raid.

Learning to control your helicraft takes some practice, as it is extremely sensitive and maneuverable. You are given five helicrafts at the beginning of the game. Each craft can absorb 100 points in damage before being destroyed, and the craft can return at any time to its carrier for a new bomb load and repairs—that is, if the carrier isn't under attack. Didn't I mention that possibility before? Well, it doesn't matter. You'll find out about this and other surprises when you play *Raid on Bungeling Bay* yourself.

Raid on Bungeling Bay
Brøderbund Software Inc.
17 Paul Drive
San Rafael, CA 94903
$29.95

Sundog: Frozen Legacy

James V. Trunzo

Requirements: Apple II-series computer with at least 64K RAM, a disk drive, and a two-button joystick.

First, take equal portions of *Star Wars* and *Star Trek*, mix in a dash of mystery and the unknown a la Infocom's *Planetfall*, and add just a hint of arcade action. Next, pour these ingredients onto a 5¼-inch floppy. Finally, boot the disk as often as you like, and enjoy for months one of the most entertaining and absorbing role-playing games on the market today.

Sundog: Frozen Legacy is a refreshingly different, graphics-oriented space game. It seems simple enough at first. You have inherited a serviceable but rundown one-man star freighter named the Sundog from a little-known uncle who has died (mysteriously). You have also inherited his obligation: to help build a colony for a religious group. You must establish the colony somewhere on the planet Jondd; find, buy, and deliver all goods needed for construction; and locate and deliver cryogenically preserved colonists who are being kept in warehouses in various cities somewhere within the Drahew Region.

But to complicate matters, you have no knowledge of trading, little knowledge of piloting the spacecraft, and the Drahew Region is *huge*. The designers of *Sundog* were ambitious, to say the least. The Drahew Region consists of 12 star systems with 18 inhabited planets—and those 18 planets are dotted with more than 900 populated cities.

Perhaps the most innovative aspect of *Sundog* is its liberal use of screen windows. The graphics range from excellent to adequate depending on which screen is in play (pilotage, tactical, on-land travel, etc.), but the windowing makes this program truly enjoyable to play.

Another nicety in *Sundog* is the variety of responses and interplay among the various people you meet. Bartenders can be very friendly when you are spending money, but curt when they feel you are wasting their time. Merchants (and even local toughs) can be bargained with, threatened, and so on, and each approach produces different results. This brings a feeling of individuality to your encounters and adds flesh to the body of the game.

Sundog allows you to save up to eight games on a disk and displays the status of each game (suspended, in play, or completed). You may, of course, delete entire games from the disk to begin new adventures. Written in *Apple Pascal 1.1*, the program handles its many sophisticated features quickly and reliably. *Sundog* is a game that can be played over and over again, even after the initial tasks have been solved.

Sundog: Frozen Legacy
FTL Games
7907 Ostrow, Suite F
San Diego, CA 92111
$49.95

Enhancements To BASIC For Atari

Tom R. Halfhill, Editor

Requirements: Atari 400/800 with 48K RAM, a disk drive, and Atari BASIC cartridge; or a 1200XL with a disk drive and Atari BASIC cartridge; or a 600XL/800XL with 64K RAM and a disk drive.

Enhancements to BASIC is an impressive utility that provides more than 40 new commands for BASIC programming. It isn't an "extended BASIC"—that is, it doesn't add any special keywords for inclusion in your programs. Rather, it's a tool kit utility that makes programming and debugging easier and more efficient. Best of all, it frees you from worrying about the infamous lockup bug which has haunted Atari BASIC programmers for years.

Before using *Enhancements to BASIC*, you must follow a 25-step setup procedure carefully described in the manual. First, *Enhancements* copies your Atari BASIC cartridge (400/800/1200XL) or built-in BASIC (600XL/800XL) into Random Access Memory (RAM). Then it modifies BASIC, adding the new commands and fixing the bug which can unpredictably lock up the Atari during program editing. Finally, *Enhancements* copies itself onto another disk for future use. From then on, you simply boot up the modified BASIC from this disk, either removing the BASIC cartridge if you have a 400, 800, or 1200XL, or holding down OPTION to disable the built-in BASIC on the 600XL/800XL.

Most of the new features are fairly standard for utilities of this type. There are commands for automatic line-numbering and renumbering, deleting blocks of lines, listing all variables, searching for lines containing a certain variable, globally changing variable names, and sending screen output to a printer. There are disk commands which lessen the need to exit BASIC to DOS: You can call disk directories, rename files, lock and unlock files, delete files, format disks, and run a machine language program at a certain address.

That would be enough for most tool kit utilities. But *Enhancements* adds some more exotic features. There are commands to restore, rename, and call directories of deleted disk files. Another command protects BASIC programs by making the listings unreadable but executable. You can lock the keyboard to prevent tampering until a personal authorization code is entered. You can change screen margins and the delay period before keys begin repeating. On a 600XL/800XL, you can change the key-repeat rate and turn the keyboard click on and off. You can display numeric values in decimal or hexadecimal (try renumbering a BASIC program in hex!). There are even programmable function keys—for example, pressing CTRL-4 automatically prints LOAD"D: or anything you like.

But perhaps the most powerful enhancement is the tracer/debugger. Most TRON/TROFF (trace on/off) utilities simply display the line numbers being executed as the program runs. *Enhancements* displays the entire line, and lets you flip back and forth between the scrolling listing and *the actual program running simultaneously on an alternate screen.* Implementing this feature on a computer with only 48K or 64K of RAM is quite impressive. Memory conflicts sometimes prevent you from tracing very large or complex programs, but nevertheless, the tracer does a remarkable job of handling such memory-juggling challenges as player/missile graphics, custom character sets, and machine language subroutines. You can also disable the tracer to conserve memory.

There are still more features we don't have room to cover, such as a handy help screen. *Enhancements* is an exceptional product.

Enhancements to BASIC
First Byte
P.O. Box 32
Rices Landing, PA 15357
$14.95 (includes shipping & handling) ©

THE BEGINNER'S PAGE

Tom R. Halfhill, Editor

FOR-NEXT Applications

Last month we covered the basics of looping with the FOR-NEXT statement. Now let's take a look at some practical applications of this essential technique.

FOR-NEXT is such a general-purpose structure, it has numerous uses. Here's an example of how you might apply it in part of a checkbook-balancing program that sums the amounts for a month's worth of checks:

```
10 PRINT "HOW MANY CHECKS
   THIS MONTH";
20 INPUT CH
30 FOR X=1 TO CH
40 PRINT "AMOUNT OF CHECK";
50 INPUT AM
60 SUM=SUM+AM
70 NEXT X
80 PRINT "TOTAL AMOUNT IS
   $";SUM
```

Let's take a careful look at this program. Try running it. Line 10 prompts the user to enter the number of checks to be added; line 20 stores the response in the variable CH. Line 30 is a little tricky. It marks the beginning of the loop with a FOR statement as shown in last month's examples, but the number of repetitions specified is a variable, not a number.

The variable, CH, contains the number of checks the user entered in response to the prompt. Therefore, the number of times the FOR-NEXT loop will repeat depends on the user's response. In effect, the program adapts itself to the user's needs. Think of what would happen if you specified the number of loops with a certain number, say 10. If the user has only 7 checks, the program would make too many loops, demanding amounts for 3 checks that were never written. If the user has 23 checks, the program would add only 10 of them together, ignoring the remaining 13.

Line 40 prompts the user to enter the dollar amount of the first check (do not type a dollar sign). Line 50 stores the response in the variable AM. Line 60 creates the variable SUM to keep track of the total and adds AM to SUM (when the first pass through the loop begins, SUM equals 0).

Line 70 marks the end of the loop. It circles back to the FOR statement on line 30. Now the loop begins its second pass. Again, a prompt asks the user to input a check amount. Again, the response is stored in the variable AM (replacing the previous amount for the first check). Again, AM is added to SUM, so SUM now contains the cumulative amounts of the first and second checks. And again, at line 70, NEXT X circles back to line 30 for the third pass.

This continues until the number of loops specified by CH is reached. Then the loop is done and the program proceeds to line 80. The program prints the total amount held in SUM and ends.

This example shows two things: first, the usefulness of FOR-NEXT loops when combined with other techniques, such as INPUT statements; and second, the flexibility of FOR-NEXT loops when modified slightly, such as specifying the number of loops with a variable instead of a particular number.

Speed READing

One of the most common applications of FOR-NEXT loops is to combine them with the READ and DATA statements. (If you aren't familiar with READ-DATA, don't worry; it's a subject for a future column.) By embedding a READ statement within a loop, you can efficiently fill an array with DATA, or POKE numbers for custom character sets or machine language subroutines into memory.

Let's try a simple example. Say you're writing some sort of calendar program that requires the computer to print a column of numbers representing the number of days in each month of the year. Without looping, you could take this approach:

```
10 PRINT "31"
20 PRINT "28"
30 PRINT "31"
40 PRINT "30"
50 PRINT "31"
60 PRINT "30"
70 PRINT "31"
80 PRINT "31"
90 PRINT "30"
100 PRINT "31"
110 PRINT "30"
120 PRINT "31"
```

But a FOR-NEXT loop with READ-DATA is much more efficient:

```
10 FOR X=1 TO 12
20 READ A
30 PRINT A
40 NEXT X
50 DATA 31,28,31,30,31,30,
   31,31,30,31,30,31
```

Line 10 sets up a FOR-NEXT loop with 12 passes (the number of elements in the DATA statement at line 50). Line 20 reads a number from the DATA statement and stores it in the variable A.

Line 30 prints the value of A, which changes after each pass through the loop. During the first pass, A equals 31 (the number of days in January) because 31 is the first DATA element. During the second pass, A equals 28 (the number of days in February) because 28 is the second DATA element. This continues for all 12 passes, concluding when A is assigned the value 31 for the twelfth month, December.

Next month, we'll continue our discussion of FOR-NEXT by showing how to put loops within loops, and even why you might want to create a loop that does *absolutely nothing*. We'll also cover some variations of FOR-NEXT in different versions of BASIC. ©

You never know who you will meet or what will be said when you're on American People/Link.

AMERICAN PEOPLE/LINK™ users meet a lot of different people. That's because PEOPLE/LINK — the nation's first all entertainment videotex network — allows anyone with a word processor, personal computer or terminal, and a modem to communicate with other users throughout the country from the privacy of their own homes.

Sometimes these electronic conversations are serious...sometimes they're outrageous...but they will always keep your interest as you talk to friends and make new ones. PEOPLE/LINK's recreational programs include:

PARTYLINE — Meet people and talk live with other PEOPLE/LINK members throughout the country in groups or privately.

CLUB-LINK — Join or start a club or group devoted to a favorite hobby, rock group, lifestyle, etc.

WHO-IS-WHO — locate other users with similar interests.

And there's much more with programs like NETMAIL, our person-to-person electronic mail, PEOPLESCAN, the national bulletin board system, and play games such as poker, blackjack, checkers, chess, backgammon, and bridge (color graphics are available for most popular computers).

And the greatest thing is that you don't have to be a computer whiz or even know how to type to use PEOPLE/LINK...just be smart enough to subscribe now!

AMERICAN PEOPLE LINK

Use your Visa, Mastercard, or American Express
Call Us Toll Free **1-800-524-0100**

Illinois Residents Call **1-312-870-5200**

American PEOPLE/LINK
Arlington Ridge Office Center
3215 N. Frontage Road—Suite 1505
Arlington Heights, IL 60004

David D Thornburg, Associate Editor

Expert Systems And The Mass Market Micro

A few months back I raised the controversial notion that, contrary to the views of the pundits, the home computer industry hadn't died. In fact (and here's where the criticism came), I argued that the true "home" computer market had yet to be born.

My point was that personal computers are unlikely to penetrate much more than 10 percent of American homes until they become as easy to use as much of the other technology we find in our homes. I agree it is silly to expect a personal computer to be as easy to use as a clothes dryer. On the other hand, why should it be any harder to use than a video disc player, especially when the video player contains more complex technology than is found in most personal computers?

Before home computers become commonplace, we must also create home software applications that will allow the computer to advance beyond the stage of being a tool for writing or record-keeping. Imagine sitting at your computer in the middle of the night and engaging in the following dialog with your home computer:

What seems to be the problem?

My child just woke up and says he itches. He has red splotches on his face.

Red splotches can indicate many things. Please call your child's doctor now. If the doctor has to call you back, please continue to answer my questions while you are waiting for the call to be returned. Are the red splotches located on other areas of the body as well?

Yes, there are some small ones on the chest.

Did your child eat any of the following foods in the last eight hours: oranges, strawberries, chocolate?

Yes, we had ham with an orange peel sauce for dinner.

Has your child ever displayed an allergic reaction to orange juice?

[And so on.]

Applying Artificial Intelligence

This hypothetical interaction could help save a family member's life, or at least reduce discomfort. Such programs—which can help solve problems ranging from diagnosing an illness to selecting the correct wine for dinner—are called *expert systems.*

Expert systems are one of the current commercial applications of research in the field of artificial intelligence. Up to this point, most expert systems have been run on fairly large computer systems, and they have been applied to massive computational tasks such as choosing the correct location for offshore oil wells.

Creating an expert system requires close interaction with human experts who are able to express their own decision-making process in terms of rules. Each rule is generally expressed in the form: *IF (conditions are true) THEN (result is likely).* Many of the more sophisticated expert systems have ways of dealing with imprecise information—assigning likelihoods to various results depending on the certainty with which the conditions are known.

The expert system program contains this set of rules (which can range in number to well over 100). The program also contains another part called an *inference engine.* The inference engine decides which rule to apply to the various information that has been entered, knows when to ask for more information, and infers a result. Once an expert system has drawn a conclusion, the user can usually enter a command such as WHY?, and the system outlines the various rules and information it used to get its result. Mathematically speaking, such programs can prove themselves.

Ferns Versus Roses

Anyone can probably list several applications for expert systems: home medicine, car problems, plant choices for the garden, choosing the right stereo, and even picking the right computer!

As home computers acquire more memory and disk storage, expert systems will become commonplace. Artificial intelligence languages such as PROLOG and LISP are now being readied for the personal computers many of us already have, so there is no technological reason that expert systems won't become a reality in our country in the next two years.

Consider that in Britain, PROLOG is available on the inexpensive Sinclair Spectrum+ for under $40. According to Sinclair, this language is selling well, and is being taught to school children who are using it to build their own expert systems. The current interest in teaching database skills in our country is a refreshing step in the right direction.

The birth of the true consumer market for computers (a market in which computers will become as commonplace as televisions) will come very soon. We who have used personal computers since the 1970s and early 1980s will be fondly remembered as the pioneers of the true information revolution.

And, to the extent that we create useful applications for these machines, we may become shapers of the revolution as well. ©

It's easy to make a copy.
It's quick.
It's illegal.
It's wrong.

It's hard to believe.

People who wouldn't think of shoplifting a software product on their lunch hour don't think twice about going back to the office and making several illegal copies of the same software.

Making unauthorized copies of software is a violation of U.S. Copyright Law. Yet, the problem has reached epidemic proportions because many people are unaware, or simply choose to ignore the law. The software industry is urging decision-makers and software users to take steps to stop software piracy in their organizations. In the meantime, the industry has been forced to prosecute willful copyright violators.

There are legal, moral and economic imperatives forbidding theft of copyrighted software.

There is a free pamphlet on the subject. Call or write for a copy. A copy. A copy. A copy for everyone you know. Please ask for Priscilla.

ADAPSO

ADAPSO
1300 North Seventeenth Street
Arlington, Virginia 22209
(703) 522-5055

Bits, Bytes, And Black Sheep

Late last fall I attended the Bits & Bytes Show at the Disneyland Hotel in Anaheim, California. Bits & Bytes was the first national computer conference for kids, and I was there to cover it for COMPUTE! and for two PBS shows—"The New Tech Times" and "Educational Computing."

The show was a terrific success—especially for children. Thousands of kids came, played with the newest computers and robots, and got a chance to tell the bigwigs of the computer industry what they thought about their products. For example, one little girl, Kimberly Williams, returned from the show and wrote to the conference organizers: "Thank you for inviting my class to the computer show. It taught me and my friends a lot about computers. The computers were very learningful to my brain."

At the show, I gave a presentation on a favorite topic of mine: the ways in which a computer could become a "sandbox" for little children. I also made a few critical remarks about the programming language Logo. I said that although I enjoy programming in Logo, I don't think computer languages are especially appropriate for younger children because the rewards are not commensurate with the amount of effort required. Also, I said that the Logo environment is somewhat artificial, abstract, and not meaningful to a small child.

Angry Reaction

I had made similar remarks at other conferences, so I didn't expect the kind of reaction I got. What a shock! Ten minutes into my talk, people in the front row rose to their feet and furiously denied that anything I had said was true. They were teachers who had been teaching Logo to their classes at school, and they said their experiences had been exactly the opposite of my own.

After listening to their point of view for a few minutes, I asked other members of the audience if they agreed. By the end of the session (which turned into a free-for-all debate), I learned that there are many different points of view about Logo and very few points of universal agreement.

However, my feelings about the Logo controversy were strengthened the other night when I picked up a copy of an excellent Canadian magazine, *Computers in Education*, and read an article by Elias Leousis, a teacher and the founder of the first full-time computer literacy program at the elementary level in the province of Quebec. In the article, entitled "Black Sheep and Logo," Leousis wrote that "Logomania" is starting to become a cult. Leousis himself uses Logo to teach programming skills, but he worries about the absurd claims made by some of Logo's admirers. "As a result of such claims," he wrote, "disillusioned educators, having followed the 'Logo route,' may cause an anti-computer backlash, destroying all advances made in the area of introducing computer literacy in the education field."

People Inside The Machine

A few years ago, I wrote a book introducing computers to children. I interviewed dozens of computer pioneers, including J. Presper Eckert, who along with John W. Mauchley invented the ENIAC, the granddaddy of today's electronic digital computers.

I wanted to call the book *The People Inside the Machine* because I concentrated on the inventors and the excitement and joy they had received from working on computers. The book showed youngsters how real people with hopes, dreams, and frailties had built computers, step by step, over many, many years. By showing the people inside the machine, I hoped to encourage young readers to see a reflection of themselves inside machines of the future. The book's message was that inventions like the computer may require a dash of genius, but even more important are hard work, a playful imagination, devotion, and stubborn, mulelike persistence in following through with your own ideas and magnificent obsessions.

As it turned out, the book was retitled *Messner's Introduction to the Computer* (Simon & Schuster, 1983), but it's still oriented to young people. If you're a grownup who wants to read about the people inside the machine, I recommend Tom Mahon's new book, *Charged Bodies: People, Power, and Politics in Silicon Valley* (NAL, paperback, 1985). Mahon's account is one of the most honest, eloquent, and fascinating books I've read in a long time. You learn about computer technology—the semiconductors, microchips, operating systems, and Winchester disks—but Mahon weaves the technology into the lives of the industry's famous and obscure pioneers, and has made what could have been a dry history of computers into a very interesting story.

Mahon doesn't pull any punches, either. He devotes equal attention to the dark side of computers as well as the light side. And he does it all in a vivid style reminiscent of Tracy Kidder's Pulitzer Prize-winning *Soul of a New Machine* (Little, Brown, 1981).

This book is an excellent primer on computer technology and the computer industry, and it will make a good computer literacy text for high school and college introductory courses on computers. ©

Fred D'Ignazio loves to get electronic mail. Here are his electronic mailboxes: The Source (BCA638); CompuServe (75166,267); MCI Mail (Fred D'Ignazio); and EasyLink (62856637).

Arlan R. Levitan

Inside XMODEM

Last month we covered the basics of uploading and downloading and how the *XMODEM protocol* developed into a system for reliably transferring files over phone lines. Now let's take a look inside XMODEM to see how it works.

XMODEM requires certain communications settings (eight data bits, no parity, one stop bit). Some terminal programs set these parameters automatically during XMODEM transfers. Others require you to set them manually.

XMODEM transfers files by sending and receiving them in blocks. Each block consists of five elements: a special character that marks the beginning of the block (called a *start of header* character); a number which identifies the sequence number of the block (incremented by one for each block sent); a number which insures the block number is correct; a 128-byte chunk of the file being transferred (the file is chopped up on the sending end and reassembled on the receiving end); and a number to verify that the 128-byte chunk of data received is the same as was sent. This number is called a *checksum*. It is the sum of the hexadecimal values of all 128 characters.

Downloading

After notifying the remote system that you want to download a certain file with XMODEM, the other system prepares the file for transmission and then waits for your signal that things are ready to begin. When you instruct your terminal program to begin the transfer, it sends an acknowledgment signal consisting of a single character called an *ACK*. The first ACK means "ready for next block."

The remote system then sends a block to your computer. Your terminal software looks for the start of header character, then checks the block number for both validity and proper sequence. Next it examines the 128 bytes of data and calculates a checksum. It compares this to the checksum actually sent by the remote system; if both are the same, it assumes the data was transmitted without errors.

As each block is successfully received, another ACK is sent. If an error is detected or the next block is not transmitted after a certain amount of time, your terminal program sends a character referred to as a *NAK* (negative acknowledgment). A NAK means, "I think we had a problem with the last block; please resend it." The transfer is aborted if a single block must be resent more than an agreed-upon number of times (usually ten). When the transfer is complete, the remote system sends a signal called *EOT* (end of transmission).

Nonstandard Protocols

Although the original standards for XMODEM file transfers are well established, not all terminal programs that advertise XMODEM capability actually conform to the standard. Here is an example of what can happen (the names have been changed to protect the innocent):

The Bitblaster computer has been available for about a year. Some kindly soul, intent upon putting far-flung Bitblaster owners in touch with each other, writes a bulletin board program for the new machine. Since he wants to encourage people to share programs for the Bitblaster, he includes an XMODEM feature. His friend writes a public domain terminal program with XMODEM capability and starts passing it around.

What they don't know is that their implementation of XMODEM was not strictly by the book. But since the terminal program and BBS work fine together, no one notices. The number of Bitblaster BBS systems rapidly multiplies, and there are thousands of copies of the free terminal program in use.

However, when Bitblaster owners start using the terminal program on other systems, they discover—much to their chagrin—that the good ol' XMODEM feature doesn't work too well on non-Bitblaster BBSs. When something like this happens, it can take years for everyone to agree on a common fix for the problem. Even commercial telecomputing packages are written by normal people, perfectly capable of making such mistakes.

Other Problems

Commercial information services are not well suited to XMODEM transfers. The protocol was developed for microcomputer-based BBSs which handle only one user at a time and can therefore devote all of their attention to the transfer. On a commercial service, hundreds of users may be logged on simultaneously, and the mainframe computer gives each user only a certain amount of attention for a certain time period.

The only way to make XMODEM work on the information services was to relax the timing standards. However, this severely limits the ability to recover from bad blocks and out-of-sync acknowledgment signals.

Although errors are relatively rare, many terminal programs and information services offer their own transfer methods in addition to (or instead of) XMODEM. There's only one hitch—often this means both computers involved in the transfer must be running the same terminal program, or in the case of information services, a terminal program written and sold by the service for your particular system.

Next month, as promised earlier, I'll show you how to save money when sending E-mail by composing your messages offline and then uploading them to the host computer. Until then, BCNU. ©

Donald B. Trivette

Escaping On A LaserJet

This Hewlett-Packard LaserJet printer is shown hooked up to an HP 150 computer, but it also works with the IBM PC and compatibles.

I got a letter last month from my friend who has a covey of computers in his spare bedroom. Five at last count! We've had a friendly rivalry going for years. I bought a pocket calculator, and he bought an Apple III. I got an IBM PC first, but he got a bigger one—the XT. I bought a PCjr, so he got a PCjr *and*, a few months later, the PC AT. Now he sends me this letter: "This is the first letter ever written with my Apple Imagewriter and Macintosh computer...it sure is fun to play with the font sizes and styles and formats...."

The font he was experimenting with looked like a cross between stencil letters and something a monk in the Middle Ages might have laboriously drawn with a quill—right justified, of course. Having run out of IBM equipment to one-up me with, he has switched to Apple. The letter isn't easy to read, but it *is* flashy. I thought about going to the Apple store and trying out a Macintosh long enough to answer his letter, but I needed something better than that—something better than IBM or Apple.

A Printer Or Copier?

It was about this time that Hewlett-Packard called to say they had a LaserJet printer for me to evaluate. Although it arrived at the airport, the LaserJet looked more like an office copying machine than an airplane. And for good reason—that's mostly what it is. The data from the computer is etched on a drum by a laser, and then transferred—one page at a time, eight pages per minute—from the drum to the paper. There's a tray for blank paper, a cartridge for toner, and a manual to show you how to fix a paper jam. Anyone who has used a copying machine will feel right at home with the LaserJet.

Connection was easy. I unplugged my Hayes modem and plugged the LaserJet cable into the modem cable. Typing two DOS commands told the PC to forget my parallel printer and talk instead to the LaserJet, which is a serial printer.

Using the LaserJet takes a little getting used to. You send something to the printer and nothing happens—at least, it doesn't look or sound like anything is happening. There's virtually no noise except for a quiet fan. And nothing immediately pops out. The printer etches the text on the drum and then waits for the page to fill up before transferring the data to paper. If you want to print a partial page, you press the form feed button. Once you've mastered the idea that printing and noise are not related, the LaserJet operates much like any other printer. (In fact, it's so much fun that I lasered a whole ream of paper playing with it.)

The print quality is superb. Although the characters are formed with tiny dots, like a dot-matrix printer, the 300 × 300 dots-per-inch resolution is so fine that you need a magnifying glass to prove how it was done. The quality of the print is better than that of my NEC 3550 printer, which has fully formed characters and uses a film ribbon. But in a duel of one-upmanship, quality isn't enough. You've got to have sizzle. The Hewlett-Packard LaserJet has plenty of sizzle.

A sample printout made with the LaserJet.

Hi David --

I got your letter and was surprised to hear that you bought a *Macintosh.* The *Imagewriter* does a fancy job of printing even if it is a little difficult to read.[1]

1 Personally, I've found this 8-point type very useful for footnotes. Does the Imagewriter support anything like this?

Custom Printing

Different type styles (fonts) are available by plugging a Read Only Memory (ROM) cartridge into the front of the printer. Each cartridge—about the same size and shape as the old eight-track audio tapes—can store up to 16 fonts, although the one I used had only eight. To select a font, you simply send a special character sequence (called an escape sequence) to the printer.

The LaserJet is more powerful than most printers, so the escape sequences are somewhat longer and more complicated. For example, to enable bold printing, the sequence on my NEC 3550 is *escE* (where *esc* represents the *escape character*, ASCII 27). On the LaserJet, the equivalent command is *esc&10Oesc(0Uesc(s1p10v0s1b5T.*

The LaserJet is a relative bargain at $3,495, considering its power and new technology, but only if you can easily take advantage of all its features without having to resort to these huge escape sequences. Fortunately, some programs are appearing which relieve you of this burden. The *Volkswriter Deluxe, Version 2.1* word processor has a printer driver especially designed for the Hewlett-Packard LaserJet printer that does a splendid job.

The accompanying figure shows part of the letter I sent to my friend to demonstrate some of the nice things the LaserJet with *Volkswriter Deluxe* can do. I didn't mention to him that the printer had to be returned to Hewlett-Packard at the end of the month. Since I haven't yet received a reply, I expect he's measuring his spare bedroom for a mainframe. ©

Bill Wilkinson

Analyzing The BASIC Bug

Last month I showed some ways to minimize the problems caused by the bugs in revision B Atari BASIC (the built-in BASIC in the 600XL and 800XL). But many of you are curious about exactly *why* these bugs happen, and what effect they can have on your programs.

Let me begin by telling what did *not* cause the error. Rev B BASIC has a peculiar problem: Each time you LOAD (or CLOAD or RUN "*filename*") a program, rev B adds 16 bytes to the size of your program. If you then save the program, the next time you load it in it grows by *another* 16 bytes, and so on.

Now believe it or else, these additional 16 bytes were put in deliberately. It seems that there is a minor, undocumented bug in the Atari S: (graphics screen) driver. Under some circumstances, it will use a few bytes below MEMHI (contents of locations $2E5–$2E6, 741–742 decimal). So, if you have a program which extends right to the very top of memory, you can wipe out a little bit of the runtime stack where GOSUBs and FORs are remembered. Somebody at the old Atari apparently had the bright idea that if BASIC told you that memory was full when your program got within 16 bytes of MEMHI, the screen/BASIC conflict could be avoided.

A Fix Gone Sour

Pretty good idea. Except for a few problems. First, BASIC doesn't save the string/array space of the runtime stack; both are created when a program is run. So the nice fact that the saved file is guaranteed to have 16 bytes of space left is negated as soon as you DIMension a string or an array or use a GOSUB or FOR. Second, the 16 bytes are added to all of BASIC's size pointers before the comparison with MEMHI is made. Good. But the newly increased value is then stored as the new pointer value. That effectively moves the program upward in memory by 16 bytes, meaning that the desired 16 bytes of free space aren't there anyway!

Well, the point of this digression is twofold: (1) This is yet another reason to use LIST and ENTER with rev B BASIC, since ENTERing a program does *not* trigger this silly 16-byte bug. (2) Several people wrote and suggested that this 16-byte bug is what causes the infamous keyboard lockup bug. Sorry, folks.

Last month, I mentioned the detailed explanation of the rev A Atari BASIC lockup bug which is to be found in COMPUTE!'s *Atari BASIC Source Book*. Well, apparently somebody at Atari read the book. Or maybe they just noticed that my company had fixed the lockup bug in one of the three or four revised versions of BASIC that we did for Atari back in 1979 (yes, that's 1979). It turns out that the lockup resulted from two missing instructions (and a total of two bytes) within the routine which "contracts" memory. (We say "contracts" because it is used when you delete a program line, so the program is contracted in size.)

Then that same somebody looked at the "expand" routine and saw almost identical code. "Aha!" they say, "Methinks there is a bug here which just hasn't been discovered yet!"

If It Ain't Broken . . .

But they were wrong. The reason the bug appeared in the contract routine is because that routine was written *after* the expand routine and copied its pattern too closely. So our unknown someone at Atari blindly added code to the healthy expand routine and introduced a very nasty new bug. In fact, because this bug appears when you add lines to an existing program, it is probably more likely to occur than the original rev A bug!

To see a demonstration of the bug, enter the following statements in direct mode (without line numbers):

```
DIM A$(249)
A$="ANY STRING YOU LIKE"
PRINT A$
PRINT A$,
PRINT A$,
```

The last two statements won't print A$ properly in either rev A or rev B Atari BASIC—in fact, they'll mess it up two different ways. Cute, eh? The problem is that tacking that comma on the end of the PRINT statement moves the string/array space (and thus A$) by one byte. Except it doesn't, really, so the variable value table address of A$ points to the wrong place in memory! Imagine your program being destroyed in a similar way. Is it any wonder you experience keyboard lockup and scrambled listings?

What former Atari employee did I bribe to get all this information about the bugs in rev B BASIC? Did I get the listing on a microdot hidden in a pack of blank disks? Sorry to disappoint you, but I did what any other hacker would do: I dragged out my trusty machine language debugger and carefully disassembled certain portions of rev B BASIC.

Finally, here's how the two bugs we have discussed were fixed in rev C BASIC, which is built into the new XE series computers (and also is available for older Ataris on cartridge at nominal cost—see last month's column). Since both bugs were caused by adding things to code which worked before, you would think that Atari could simply take the "fixes" back out. Nope. Instead, they patched over the extraneous instructions with what are effectively NOP (NO oPeration) instructions. Tacky? Well, I've certainly done it to DOS here in this column enough times, so who am I to say? ©

C Regena

Multiple Choice Test

I've seen a number of computer programs written for multiple choice tests. The computer is an ideal way to administer such tests because it can mix up the test questions so each run is different. However, all the programs I have seen always print the choices in the same order. This month's program is a general-purpose multiple choice tester that randomly arranges both the questions (without repetition) *and* the possible answers.

This program can be used for questions on any topic. Computer literacy questions are included here for an example.

The questions and answers are in DATA statements. Each DATA statement contains six items. The first item is the question; the next four are the possible answers; and the last item is the number of the correct answer. The final DATA statement signals the end of the question list:

```
1350 DATA ZZZ,Z,Z,Z,Z,0
```

You may use any number of possible questions that will fit in the computer's memory. Line 190 is a DIMension statement that allows for 30 possible questions. To increase the number of questions, change all the 30s in line 190.

Reading The Data

Line 160 defines the number of questions in the test, and line 300 performs the procedure for the specified number of questions. If you want to alter this number, change the 20 in these two lines plus the printed score in line 690. Also make sure you have as many or more questions and answers in the DATA statements as you want in the test.

The variable I is used as a counter for the questions. Questions read from the DATA statements are stored in the string variable T$, the four possible answers are stored in A$, and the number of the correct answer is stored in B. These values are in arrays to keep the answers with the corresponding questions.

As the information is being read in, S$(I) is set equal to A for use as a signal so questions won't be repeated during the quiz. When a question X is printed, S$(X) is set equal to "" (null). Line 320 chooses a random number X, but if S$(X) is null, the question has previously been used and a different X must be chosen. Line 350 prints the question.

Lines 370–390 define C(J) for the four answers to mix up the order in which the answers are printed. Line 400 randomly chooses D for the correct answer. The C variable for the correct answer is set to zero so it cannot be used in another position. Lines 430–490 mix up the order of the answers, making sure the correct answer is in the right position and each answer is used only once. Lines 500–530 print the four answers with the possible choices A, B, C, and D.

Lines 540–580 receive the student's answer, making sure it is a letter from A to D, and then print the choice. Line 590 checks to see if the key pressed is the correct choice. Line 600 prints the message for an incorrect answer, then prints the correct answer. Line 620 prints CORRECT for a correct answer, and line 630 increments the score, SC. Lines 640–670 wait for the student to press ENTER before going to the next question.

Lines 680–700 clear the screen, then print the score.

To customize the test, simply change the questions and answers in the DATA statements, making sure you have enough questions for a complete quiz and that the last DATA statement contains ZZZ to signal the end. You might also prefer a fancier title screen.

Here is an example of changing the DATA statements. Suppose your question is "In which year did Columbus discover America?" with the possible answers 1256, 1492, 1776, and 1812. The correct answer is in the second position. The DATA statement would look like this:

```
720 DATA IN WHICH YEAR DID
COLUMBUS DISCOVER AMERICA?
730 DATA 1256,1492,1776,1812,2
```

If you want to save typing effort, you can obtain a copy of this program by sending a cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Please be sure to specify the name of the program and that you need the TI version.

Multiple Choice Test

```
100 REM  MULTIPLE CHOICE TE
    ST(5 SPACES)
110 CALL CLEAR
120 PRINT "**********************
    **********"
130 PRINT "*  MULTIPLE CHOI
    CE TEST  *"
140 PRINT "**********************
    **********"
150 PRINT ::::::
160 PRINT "TEST OF 20 QUEST
    IONS"
170 PRINT ::"PRESS LETTER O
    F CORRECT"
180 PRINT :"ANSWER FOR EACH
     QUESTION."
190 DIM T$(30),A$(30,4),B(3
    0),S$(30),AA$(4)
200 I=1
210 READ T$(I),A$(I,1),A$(I
    ,2),A$(I,3),A$(I,4),B(I
    )
220 IF T$(I)="ZZZ" THEN 260
230 S$(I)="A"
240 I=I+1
250 GOTO 210
260 I=I-1
270 PRINT ::"PRESS <ENTER>
    TO START."
```

```
280 CALL KEY(0,K,S)
290 IF K<>13 THEN 280
300 FOR P=1 TO 20
310 RANDOMIZE
320 X=INT(I*RND)+1
330 IF S$(X)="" THEN 320
340 CALL CLEAR
350 PRINT T$(X)::
360 S$(X)=""
370 FOR J=1 TO 4
380 C(J)=1
390 NEXT J
400 O=INT(4*RND)+1
410 AA$(O)=A$(X,B(X))
420 C(B(X))=0
430 FOR J=1 TO 4
440 IF J=O THEN 490
450 E=INT(4*RND)+1
460 IF C(E)=0 THEN 450
470 AA$(J)=A$(X,E)
480 C(E)=0
490 NEXT J
500 FOR J=1 TO 4
510 PRINT CHR$(64+J);". ";A
    A$(J)
520 NEXT J
530 PRINT ::
540 CALL SOUND(100,1497,2)
550 CALL KEY(0,K,S)
560 IF (K<65)+(K>68)THEN 55
    0
570 CALL HCHAR(23,3,K)
580 PRINT
590 IF K=64+O THEN 620
600 PRINT "NO, THE ANSWER I
    S ";CHR$(64+O);"."
610 GOTO 640
620 PRINT "CORRECT"
630 SC=SC+1
640 PRINT :"PRESS <ENTER>."
650 CALL KEY(0,K,S)
660 IF K<>13 THEN 650
670 NEXT P
680 CALL CLEAR
690 PRINT "OUT OF 20 QUESTI
    ONS,"
700 PRINT "YOUR SCORE IS ";
    SC::::
710 GOTO 1360
720 DATA ONE OF THE MAJOR A
    TTRACTIONSOF A COMPUTER
     IS THAT IT
730 DATA HAS ACTIVE INVOLVE
    MENT.,IS EXPENSIVE.,IS
    A STATUS SYMBOL.
740 DATA ALLOWS UNINVOLVEME
    NT.,1
750 DATA A VIDEO GAME IS BE
    ST(8 SPACES)DESCRIBED A
    S A(N)
760 DATA EXPENSIVE TOY.,SPE
    CIAL PURPOSE COMPUTER.,
    HOME COMPUTER.,EDUCATIO
    NAL TOY.,2
770 DATA THE COMPUTER OWES
    ITS(7 SPACES)FLEXIBILIT
    Y TO THE FACT THATIT IS
780 DATA SMALL.,COMPLICATED
    .,PROGRAMMABLE.,AN ELEC
    TRONIC DEVICE.,3
790 DATA "BECAUSE A COMPUTE
    R IS(7 SPACES)PROGRAMMA
    BLE,"
800 DATA IT CAN BE USED TO
    PERFORM(3 SPACES)ONLY A
     LIMITED NUMBER OF
    (4 SPACES)FUNCTIONS.
810 DATA IT CANNOT BE USED
    FOR(7 SPACES)EDUCATIONA
    L PURPOSES.
820 DATA IT CANNOT BE USED
    FOR(7 SPACES)ENTERTAINM
    ENT.
830 DATA IT CAN BECOME A GE
    NERAL(5 SPACES)PURPOSE
    TOOL.,4
840 DATA THE MAIN ADVANTAGE
     OF A(5 SPACES)COMPUTER
     AS OPPOSED TO
    (6 SPACES)OTHER CALCULA
    TING DEVICES(3 SPACES)I
    S ITS
850 DATA COST.,SIZE.,PORTAB
    ILITY.,PROGRAMMABLE NAT
    URE.,4
860 DATA BOOKS AND MANUALS
    THAT(6 SPACES)ACCOMPANY
     A COMPUTER-RELATEDPROD
    UCT ARE
870 DATA SOFTWARE.,DOCUMENT
    ATION.,DATA.,COMPU-FORM
    S.,2
880 DATA VISICALC IS BEST D
    ESCRIBED  AS A(N)
890 DATA TUTORIAL PROGRAM.,
    ELECTRONIC SPREADSHEET.
900 DATA EDUCATIONAL PROGRA
    M.,ENTERTAINMENT PROGRA
    M.,2
910 DATA ALL OF THE FOLLOWI
    NG ARE(4 SPACES)PROGRAM
    MING LANGUAGES EXCEPT
920 DATA BASIC.,PASCAL.,VIS
    ICALC.,LOGO.,3
930 DATA ONE OF THE MAJOR P
    ROBLEMS INACQUIRING COM
    PUTER LITERACY IS
940 DATA PEOPLE NEED TO BE
    SKILLED(3 SPACES)IN MAT
    HEMATICS TO USE
    (7 SPACES)COMPUTERS.
950 DATA THE COMPUTER IS A
    VERY(6 SPACES)COMPLICAT
    ED MACHINE.
960 DATA THE FIELD HAS ITS
    OWN(7 SPACES)LEXICON OR
     LANGUAGE.
970 DATA PEOPLE NEED A BACK
    GROUND(4 SPACES)IN LOGI
    C AND STATISTICS.,3
980 DATA THE PARTS OF A COM
    PUTER ARE ARRANGED IN S
    UCH A WAY AS TOFORM A(N
    )
990 DATA SYSTEM.,MACHINE.,S
    UBSYSTEM.,ORGANIZATION.
    ,1
1000 DATA THE PROCESSING OF
     DATA IN A COMPUTER SY
    STEM RESULTS IN  THE G
    ENERATION OF
1010 DATA A PROGRAM.,READOU
    TS.,INFORMATION.,STATI
    STICS.,3
1020 DATA "BASICALLY, A COM
    PUTER IS(4 SPACES)INTE
    NDED TO PRODUCE"
1030 DATA INFORMATION.,DATA
    .,STATISTICS.,PROGRAMS
    .,1
1040 DATA THE BASIC FUNCTIO
    N OF A(5 SPACES)COMPUT
    ER IS TO TRANSFORM
1050 DATA PROGRAMS INTO DAT
    A.
1060 DATA DATA INTO PROGRAM
    S.
1070 DATA INFORMATION INTO
    DATA.
1080 DATA DATA INTO INFORMA
    TION.,4
1090 DATA "BY USING A -----
    -, ONE MAY  CONNECT A
    COMPUTER TO THE
    (3 SPACES)TELEPHONE TO
     PERMIT COMPUTERCONFER
    ENCING."
1100 DATA ADAPTER,CONNECTOR
    ,CONFERENCE LINK,MODEM
    ,4
1110 DATA INTANGIBILITY IS
    A MAJOR(4 SPACES)CHARA
    CTERISTIC OF
1120 DATA SOFTWARE.,THE COM
    PUTER.,HARDWARE.,MAGNE
    TIC DISKS.,1
1130 DATA THE USE TO WHICH
    A COMPUTER IS PUT IS C
    ALLED A(N)
1140 DATA PROGRAM.,ROUTINE.
    ,APPLICATION.,FUNCTION
    .,3
1150 DATA INSIDE THE COMPUT
    ER(9 SPACES)INFORMATIO
    N IS REPRESENTED  BY
1160 DATA PUNCHED CARDS.,EL
    ECTRONIC SIGNALS.,MAGN
    ETIC TAPE.,MAGNETIC DI
    SKS.,2
1170 DATA THE ON/OFF PATTER
    N THAT IS  USED IN THE
     COMPUTER IS THE BASIS
     OF THE
1180 DATA CIRCUIT CODE.,BIN
    ARY CODE.,BINOMIAL COD
    E.,BIDECIMAL CODE.,2
1190 DATA "WITH TELECOMMUTI
    NG,(9 SPACES)INFORMATI
    ON IS MOST COMMONLYTRA
    NSMITTED BETWEEN
    (9 SPACES)TERMINALS"
1200 DATA BY RADIO.,OVER TE
    LEPHONE WIRES.
1210 DATA VIA SATELLITE.,BY
     TELEVISION.,2
1220 DATA A COMPUTER PROGRA
    M IS AN(4 SPACES)EXAMP
    LE OF
1230 DATA HARDWARE.,SOFTWAR
    E.,FIRMWARE.,FLEXWARE.
    ,2
1240 DATA THE FIRST ELECTRO
    NIC(8 SPACES)COMPUTER
    WAS
1250 DATA ENIAC.,ENID.,IBM
    MARK I.,IBM CYBERNAUGH
    T.,1
1260 DATA THE COMPUTER IS I
    NSTRUCTED  OR TOLD WHA
    T TO DO BY
1270 DATA HARDWARE.,FIRMWAR
    E.,SOFTWARE.,SMARTWARE
    .,3
1280 DATA THE MOST SIGNIFIC
    ANT FACTOR IN PURCHASI
    NG A COMPUTER IS
1290 DATA RELATIVE COST.,AV
    AILABLE SOFTWARE.,AVAI
    LABLE HARDWARE.,AVAILA
    BLE FIRMWARE.,2
1300 DATA WHICH IS THE MOST
     COMMON(4 SPACES)TYPE
    OF SECONDARY STORAGE
    (3 SPACES)CURRENTLY US
    ED IN PERSONAL  COMPUT
    ERS?
1310 DATA FLOPPY DISKS,BUBB
    LE MEMORY,ELECTRIC CON
    DUCTORS,TUNNEL JUNCTIO
    N MEMORY,1
1320 DATA RAM IS USED AS A
    MEASURE OF
1330 DATA PRIMARY STORAGE C
    APACITY.,PROCESSING PO
    WER.
1340 DATA PROCESSING SPEED.
    ,WORD LENGTH.,1
1350 DATA ZZZ,Z,Z,Z,Z,0
1360 END
```

Housepainter
Inverted Video On The Commodore 64

Jim Butterfield, Associate Editor

By fooling the eye with reverse characters, you can make a single-colored object appear to pass over a multicolored background—without sprites or machine language. The technique can also be adapted to the VIC-20, Plus/4, and 16.

If you've ever played around with video effects on the Commodore 64, you know that ordinary video involves one background color and a choice of printing colors. In other words, you can print green, red, and yellow on a white background, but you can't go the other way and print, say, white on a multicolored background. However, you can use an easy trick to get the same effect.

Here's the objective: Assume you want to move an object over a multicolored background. Maybe it's a small black bug flying over terrain which is green (grass), blue (water), and white (ice). You don't want the bug to change color as it passes these areas; it must stay black. Yet the 64 seems to be set up to work the opposite way. The foreground color may change from one character cell to another, but the background must be one solid color across the whole screen.

There are many special features you could use to get around this problem, of course. Extended color mode allows you a choice of backgrounds; it's attractive for many uses and too little used by most programmers. Sprites may be placed anywhere on the screen over existing character patterns, giving a color-over-color effect. You could even use a split-screen technique to create multiple backgrounds.

But you can also achieve the effect without resorting to special features. To invert the screen effects, we must invert our thinking.

Holes In The Sky

When I was a small child, I used to think that stars were tiny holes in a black curtain that covered the night sky. In other words, they were points of light shining through. I couldn't tell the difference between a white light source set against a black background and a white background shining through gaps in a black foreground.

That's the technique we'll use here. Since the 64 has one background color and a choice of many foreground colors, let's flip them over to create one "foreground" color against a mixture of many "background" colors. We'll have to work in reverse characters to switch background and foreground, but that's no hardship.

Let's try a simple example program which moves a white letter A over a multicolored background:

```
100 PRINT CHR$(144);CHR$(147)
110 POKE 53281,1
```

This clears the screen (147) and sets the colors—background to white (53281) and cursor to black (144).

```
120 FOR J=0 TO 39
130 POKE 1024+J,160
140 POKE 55296+J,RND(1)*14+2
150 NEXT J
```

This POKEs reverse spaces (160) into the top line of screen memory, and puts random colors into color memory beginning at 55296. Note that we need *reverse* space characters. Ordinary spaces would show the background color only, but the reversed ones show the random foreground colors. These colors will *seem* to be background, because they won't change.

```
160 FOR J=0 TO 38
170 POKE 1024+J,160
180 POKE 1025+J,129
190 FOR K=1 TO 100
200 NEXT K,J
```

These final lines move the white letter A from left to right across the top of the screen. The screen code for A is a value of 1, but the program adds 128 to get a value of 129 for a reverse A. As the A moves to its new position, it is erased from its old spot by POKEing a reverse space there (160).

An Optical Illusion

We know the program moves a reverse A along the top line of the screen. And we know that each character is a different, random, color as we print it. But when we enter RUN and watch the program work, we see a white A moving across the top. Why? Because we're really seeing the white background color "peeking through" the reversed foreground. The optical illusion is complete: A single foreground color seems to be moving across a multicolored background.

Let's use this principle in a short program called "Housepainter." It's part game, part puzzle, and part coordination exercise. You have five minutes to paint the house, using the four special function keys to move the brush. You are not allowed to paint over any area twice, so be careful and plan your work. By the way, it *is* possible to succeed, although you may have a lot of trouble at first.

The program is entirely in BASIC, which accounts for the delay

We wrote the book on the...

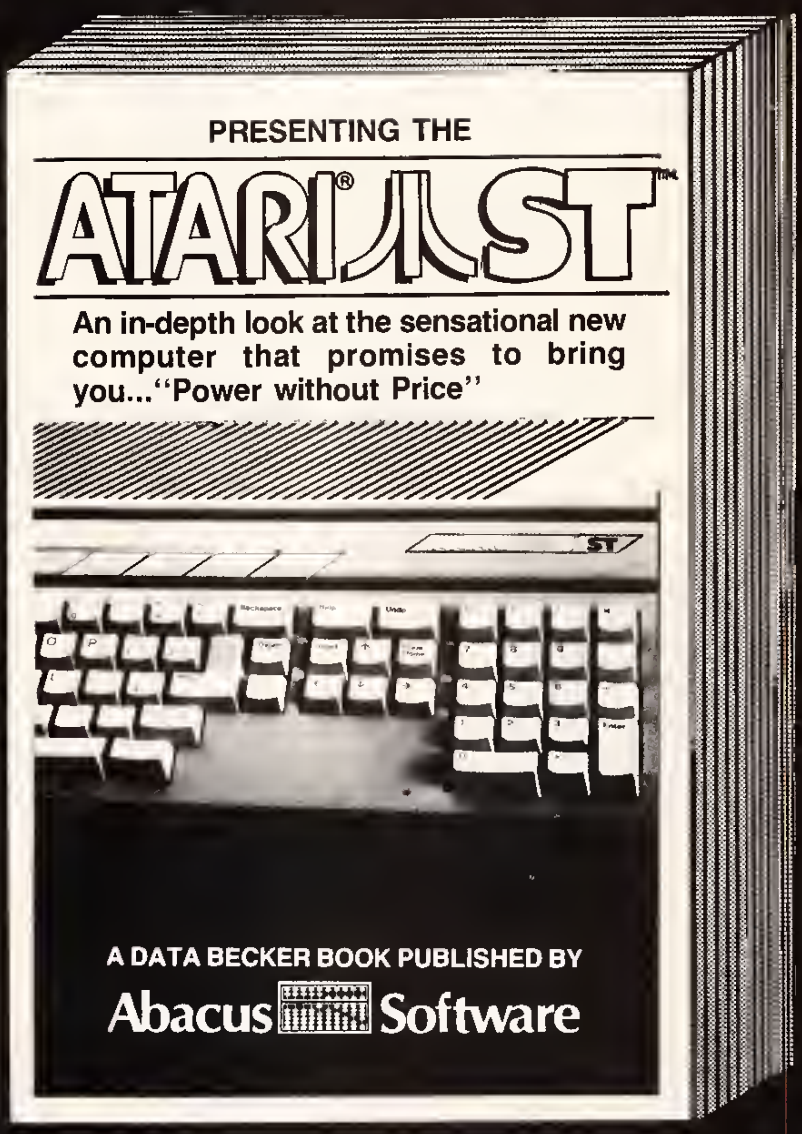
ATARI® ST™

First there was the fabulously successful VIC-20. Then came the record-breaking Commodore-64.

Now Jack Tramiel has launched his third home computer, the ATARI ST.

The ST promises to shatter all existing price-performance barriers and become a leader in the home-computer market.

This book, **PRESENTING THE ATARI ST** gives you an in-depth look at this sensational new computer that promises to bring you... "Power without the Price." Some of the topics include: • History of Atari • Overview of the ST • Construction, Operating System • Peripherals • Languages • User Programs, and more.



Price $16.95

Available June 1st Call for the name of your nearest dealer

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 For Fast Service Call (616) 241-5510

as it sets up the screen. The house is drawn with several rectangles defined in DATA statements. Note that the characters showing the time in the upper left of the screen seem normal, but are really in reverse video. The white is the background color peeking through. Similarly, the white paint itself and the circular brush are really background colors. That's how the white brush can move inside a red shed and across a yellow house: It's inverted video.

Housepainter

Please refer to "COMPUTE!'s Guide to Typing in Programs" before entering this listing

```
100 PRINT "{CLR}{DOWN}HOUSEPAI
    NTER{2 SPACES}-{2 SPACES}J
    IM BUTTERFIELD"    :rem 228
110 PRINT "{3 SPACES}TRY TO PA
    INT THE HOUSE USING THE
                       :rem 43
120 PRINT "FUNCTION KEYS TO MO
    VE THE BRUSH.      :rem 81
130 PRINT "{3 SPACES}YOU'RE NO
    T ALLOWED TO PAINT OVER
                       :rem 82
140 PRINT "AN AREA ... SO DON'
    T PAINT YOURSELF   :rem 105
150 PRINT "INTO A CORNER.
                       :rem 183
160 PRINT "{3 SPACES}YOU HAVE
    {SPACE}FIVE MINUTES.
    {2 SPACES}LEAVE THE:rem 50
170 PRINT "PAINT BRUSH IN THE
    {SPACE}RED SHED WHEN
                       :rem 240
180 PRINT "YOU'RE FINISHED.
    {2 SPACES}PRESS ANY KEY TO
     GO.               :rem 64
190 FOR J=1 TO 1E4:GET X$:IF X
    $="" THEN NEXT J   :rem 154
200 DATA 3,0,24,0,39   :rem 193
210 DATA 7,10,20,10,30 :rem 27
220 DATA 7,8,9,15,18   :rem 213
230 DATA 3,15,17,13,15 :rem 42
240 DATA 3,16,18,25,27 :rem 51
250 DATA 3,17,20,19,21 :rem 43
260 DATA 2,18,20,31,33 :rem 41
270 DATA -1            :rem 17
300 PRINT CHR$(159);CHR$(147);
    "WAIT";CHR$(142);CHR$(8):P
    OKE 53281,1        :rem 160
310 C0=55296:S0=1024   :rem 16
320 REM                :rem 121
330 READ C:IF C<0 GOTO 400
                       :rem 58
340 READ V1,V2,H1,H2   :rem 57
350 FOR J=V1*40 TO V2*40 STEP
    {SPACE}40          :rem 116
360 FOR X=C0+J+H1 TO C0+J+H2
                       :rem 209
370 POKE X,C           :rem 144
380 NEXT X,J           :rem 168
390 GOTO 330           :rem 107
400 FOR J=S0+4 TO S0+999:POKE
    {SPACE}J,160:NEXT J:rem 25
410 FOR J=C0 TO C0+999 :rem 98
420 IF (PEEK(J)AND15)=7 THEN N
    =N+1               :rem 71
430 NEXT J             :rem 32
440 TI$="000000"       :rem 250
450 V=19:H=32          :rem 186
460 POKE V0*40+H0+S0,160
                       :rem 241
470 POKE V*40+H+S0,209:rem 150
480 V0=V:H0=H          :rem 236
490 PRINT CHR$(19);CHR$(18);TI
    $                  :rem 216
500 IF TI$>"000499" GOTO 680
                       :rem 116
510 K=PEEK(203)        :rem 41
520 GET K$:K1=ASC(K$+CHR$(0))
                       :rem 29
530 IF K1=133 THEN V1=-1
                       :rem 200
540 IF K1=134 THEN H1=-1
                       :rem 188
550 IF K1=135 THEN H1=1
                       :rem 145
560 IF K1=136 THEN V1=1
                       :rem 161
570 IF K<64 GOTO 590   :rem 243
580 V1=0:H1=0          :rem 177
590 V=V0+V1:H=H0+H1    :rem 68
600 P=C0+V*40+H:IF P=P9 GOTO 4
    90                 :rem 205
610 C=PEEK(P)AND15     :rem 22
620 IF C<>7 AND C<>2 GOTO 490
                       :rem 179
630 IF C9=7 THEN POKE P9,1:N=N
    -1                 :rem 205
640 P9=P:C9=C          :rem 230
650 IF C<>2 OR N>0 GOTO 460
                       :rem 73
660 PRINT " WINNER! "  :rem 97
670 END                :rem 116
680 PRINT " YOU MISSED";STR$(N
    );" SQUARES"       :rem 203
```

Break the BASIC language barrier

VIDEO BASIC-64 - ADD 50+ graphic and sound commands to your programs with this super development package. You can distribute free RUN-TIME version without paying royalties!
ISBN# 0-916439-26-7 **$59.95**

BASIC COMPILER 64 - compiles the complete BASIC language into either fast 6510 machine language and/or compact speedcode. Get your programs into high gear and protect them by compiling.
ISBN# 0-916439-17-8 **$39.95**

MASTER-64 - professional development package for serious applications. Indexed file system, full screen management, programmer's aid, BASIC extensions, 100 commands.
ISBN# 0-916439-21-6 **$39.95**

PASCAL-64 - full Pascal with extensions for graphics, sprites, file management, more. Compiles to 6510 machine code and can link to Assembler/Monitor routines.
ISBN# 0-916439-10-0 **$39.95**

ADA TRAINING COURSE - teaches you the language of the future. Comprehensive subset of the language, editor, syntax checker/compiler, assembler, disassembler, 120+ page guide.
ISBN# 0-916439-15-1 **$59.95**

FORTH-64 - loaded with hires graphics, complete synthesizer control, full screen editor, programming tools, assembler.
ISBN 0-916439-32-1 **$39.95**

C LANGUAGE COMPILER - a full C language compiler. Conforms to the Kernighan & Ritchie standard, but without bit fields. Package includes editor, compiler and linker.
ISBN# 0-916439-28-3 **$79.95**

ASSEMBLER MONITOR-64 - a macro assembler and extended monitor package. Assembler supports floating point constants. Monitor supports bank switching, quick trace, single step, more.
ISBN# 0-916439-11-9 **$39.95**

XREF-64 - indispensible tool for BASIC programmer cross-references all references to variable and line numbers.
ISBN# 0-916439-27-5 **$17.95**

OTHER TITLES ALSO AVAILABLE - WRITE OR CALL FOR A FREE COMPLETE CATALOG
Call today for the name and address of your nearest local dealer.

PHONE: (616) 241-5510

For postage and handling include $4.00 ($8.00 foreign) per order. Money order and checks in U.S. dollars only. Mastercard, VISA and American Express accepted. Michigan residents incl 4% sales tax.

FREE PEEKS & POKES WALL POSTER INCLUDED WITH EVERY SOFTWARE PURCHASE

You Can Count On

Abacus Software

P.O. Box 7211 Grand Rapids, MI 49510 - Telex 709-101 - Phone 616/241-5510

BASIC File Editor For Commodore

Henry A. Doenlen

Edit ASCII files in the form of numbered BASIC lines with this short utility for the Commodore 64 and VIC-20.

One of the best features of any Commodore computer is its BASIC line editor. By using the insert, delete, and cursor control keys, you can easily move the cursor anywhere on the screen and edit a BASIC program without having to retype entire lines.

Unfortunately, ASCII data files—files of characters such as those produced by many word processor or database management programs—are not so easy to edit. To change one data item in a file, you must either rerun the program that produced the file, or write another program that reads the file, makes the change you want, and writes the file back to disk or tape. Both options are time-consuming.

Disguising ASCII As BASIC

Although the Commodore BASIC editor is not designed to edit such files, you can make it do the job with a simple trick: Disguise the ASCII data as a collection of BASIC lines by adding line numbers and quotation marks. For example, enter this line:

```
10 "I'M REALLY ASCII DATA"
```

You can't execute this line in a program, of course, without getting an error. But the BASIC editor can handle it with ease, letting you add or delete characters, change the line number, or relist the line. "BASIC File Editor" uses this trick to let you edit ASCII files, adding artificial line numbers and quotation marks when it loads a file into memory, and deleting them when the file is resaved.

Type in BASIC File Editor and save it. If you are using a Commodore 64 or VIC-20 with a disk drive, enter the program exactly as it is listed.Tape users should omit lines 5, 220, and 360–387, change the 42 in line 350 to 35, and change the following lines as shown:

```
110 PRINT"{DOWN}LOAD FILE":INP
    UTF$                   :rem 232
120 CLOSE1:OPEN1,1,0,F$:rem 38
230 CLOSE1:OPEN1,1,1,F$:rem 41
```

Before editing an important file, you may want to practice on a test file. This five-line program makes a test file for you (tape users should change the 8 in line 10 to a 1):

```
10 OPEN1,8,1,"@0:ASCTEST,P,W"
20 PRINT#1,"THIS PROGRAM MAKES
   "
30 PRINT#1,"ASCII DATA LOOK"
40 PRINT#1,"LIKE BASIC LINES."
50 CLOSE1
```

After making the test file, load and run BASIC File Editor. The menu offers four options: You can press L to load a file, S to save a file, C to clear data from memory, or E to list your data and exit the program. To load the test file, press L, then type ASCTEST when prompted for the filename. (Disk users should then enter P to indicate that ASCTEST is stored as a program file, as indicated by the ,P in line 10.) After the main menu reappears, press E to list the data and return to immediate mode.

Editing Your Data

As you can see, the ASCTEST data is spliced onto the end of BASIC File Editor in the form of numbered program lines. The data lines can be edited like any other BASIC lines. Try making some changes. To add new data at the end of the file, enter new lines with higher line numbers. Use intermediate line numbers to insert new data between existing lines. Don't forget that the data in each line must be enclosed in quotation marks.

When you're done editing, enter RUN to reenter the File Editor program and press S to resave the file. Again you'll be prompted to enter a filename. If you are using a disk drive, do *not* use the same filename (ASCTEST) unless you want the revised file to replace the original file. If you use a different filename, a new file is created. Of course, if you are using tape you must also be careful to rewind the cassette if you want the old file erased, or be careful not to overwrite the old copy if you want it preserved.

Now press C to clear the data area, and L to load your new file. After it loads, press E to verify that the changes were successful. If you had not pressed C, the new file would have been appended to the data already in memory. This makes it easy to append one file to another.

You should avoid using RUN/STOP or RUN/STOP–RESTORE to break out of the program. Always exit by pressing E at the menu, or important memory pointers will be left scrambled. If you do break out, rerun the program, then do a load followed by a clear before attempting any further editing.

When A Program Isn't A Program

For tape users, there's only one simple way to store ASCII data: as a tape data file. Hence, the tape version of the File Editor program works only with such files. However, disk users have greater flexibility in choosing a file type.

The most common format for character data storage on disk is the sequential file. Such files are easy to create: Simply OPEN 1,8,1, "*filename*, S,W" (the final ,S,W indicates that the file called *filename* is to be a Sequential file open for Writing). Then use PRINT#1 to write the desired data to the file and CLOSE 1 when finished. Such files will show up in the disk directory as SEQ. However, it's also possible to store the data in a program file. The pro-

cedure is the same as for sequential files, except that the ,S,W in the OPEN statement is changed to ,P,W (where the P indicates a program file). Otherwise, you still use PRINT# to write data to the file, as illustrated in the ASCTEST file created in the example above. There are several advantages to storing data in program file format. For example, with careful planning the program file of ASCII data can be retrieved with LOAD, which is significantly faster than using GET# or INPUT#. You may only rarely encounter ASCII data stored in program files, but the File Editor program can handle it in case you do.

Although BASIC File Editor allows you to edit ASCII data stored in program files, it does not allow you to edit BASIC *programs* stored in program files. While the file type is the same, all the BASIC keywords in a program are represented in the disk program file as single characters called tokens, which appear as reverse-video characters. Moreover, changes which affect the length of any program lines will cause the edited program to crash when loaded and run. Also, the File Editor can only be used to edit ASCII data files, which means it is not directly compatible with some database and word processing programs—including COMPUTE!'s *SpeedScript* word processor—which store characters as their Commodore screen code values rather than as ASCII values. (*SpeedScript* does allow you to print an ASCII file to disk, which could then be edited.)

It is possible to use the File Editor to load ASCII data from program files and store it into sequential files, and vice versa. However, in this case the replace feature will not function properly. That is, you cannot use the same filename for the edited file if you are storing it as a different type from the original.

Customizing The Editor

BASIC File Editor works best with ordinary alphanumeric data (letters and numbers), such as you might find in a word processing file. Carriage return characters (CHR$(13)) are interpreted as separators. When BASIC File Editor finds a carriage return while loading the file, it terminates the current data line and begins a new one.

It should not be difficult to customize this program for your own particular needs. Before doing so, however, look at line 350. The FOR-NEXT statement sets up a loop that counts through the lines of the program (42 for disk or 35 for tape). If you add or delete any lines in the File Editor program, you'll need to change this value from 35 or 42 to whatever is appropriate.

BASIC File Editor

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing

```
5 OPEN15,8,15          :rem 195
10 PRINT "{CLR}{3 SPACES}BASIC
   FILE EDITOR {DOWN}":rem 31
20 PRINT "{HOME}{2 DOWN}SELECT
   OPTION:"            :rem 59
30 PRINT " L=LOAD FILE":PRINT"
   S=SAVE FILE":PRINT" C=CLEA
   R":PRINT" E=EXIT/LIST"
                       :rem 11
40 GETA$:IFA$=""THEN40:rem 235
50 IFA$="E"THENCLOSE15:LIST390
   -:END               :rem 122
60 FORC=1TO3:IFA$=MID$("LSC",C
   ,1)THENONCGOSUB80,210,300:G
   OTO10               :rem 201
70 NEXTC:GOTO20        :rem 190
80 POKE45,PEEK(55):POKE46,PEEK
   (56)-1:GOSUB340:L=PEEK(44)*
   256+PEEK(43)        :rem 11
90 IFPEEK(L)+PEEK(L+1)=0THEN11
   0                   :rem 49
100 N=PEEK(L+3)*256+PEEK(L+2):
    L=PEEK(L+1)*256+PEEK(L):GO
    TO90               :rem 190
110 PRINT"{DOWN}LOAD FILE":INP
    UTF$:PRINT"{DOWN}PROGRAM O
    R SEQUENTIAL{2 SPACES}(P/S
    )":INPUTT$         :rem 29
120 CLOSE1:OPEN1,8,2,F$+","+T$
    +",R":GOSUB360:IFFL=1THEN1
    0                  :rem 6
130 A=L:T=PEEK(46)*256+PEEK(45
    )-10               :rem 104
140 L=A:A=A+3:N=N+10:POKEA,N/2
    56:POKEA-1,N-PEEK(A)*256:A
    =A+1:POKE A,34     :rem 4
150 A=A+1:GET#1,C$:D=ASC(C$+CH
    R$(0)):IF D<>13ANDA<TTHENP
    OKEA,D:GOTO150     :rem 80
160 POKEA,34:POKEA+1,0:A=A+2:P
    OKEL+1,A/256:POKEL,A-PEEK(
    L+1)*256           :rem 208
170 IFST=0ANDA<TTHEN140
                       :rem 163
180 IFA>=TTHENPRINT"OUT OF MEM
    ORY"               :rem 158
190 POKEA,0:POKEA+1,0:A=A+2:PO
    KE4,A/256:POKE 3,A-PEEK(4)
    *256               :rem 155
200 GOSUB330:CLOSE 1:RETURN
                       :rem 163
210 GOSUB350:PRINT"{DOWN}SAVE
    {SPACE}FILE":INPUTF$
                       :rem 74
220 PRINT"{DOWN}PROGRAM OR SEQ
    UENTIAL{2 SPACES}(P/S)":IN
    PUTT$              :rem 143
230 CLOSE1:OPEN1,8,2,"@0:"+F$+
    ","+T$+",W":GOSUB360:IFFL=
    1THEN10            :rem 38
240 IFPEEK(A)+PEEK(A+1)=0THENC
    LOSE1:RETURN       :rem 119
250 A=A+3              :rem 180
260 A=A+1:IFPEEK(A)=0THENPRINT
    #1,"":A=A+1:GOTO240
                       :rem 126
270 IFPEEK(A)<>34THEN260
                       :rem 135
280 A=A+1:IFPEEK(A)=34THEN260
                       :rem 160
290 PRINT#1,CHR$(PEEK(A)):GOT
    O280               :rem 249
300 PRINT"CLEAR":GOSUB350:POKE
    A,0:POKEA+1,0:A=A+2
                       :rem 219
310 POKE4,A/256:POKE3,A-PEEK(4
    )*256              :rem 215
320 GOSUB330:RETURN    :rem 197
330 POKE46,PEEK(4):POKE45,PEEK
    (3)                :rem 172
340 POKE47,PEEK(45):POKE48,PEE
    K(46):POKE49,PEEK(45):POKE
    50,PEEK(46):RETURN:rem 242
350 A=PEEK(44)*256+PEEK(43):FO
    RN=1TO42:A=PEEK(A+1)*256+P
    EEK(A):NEXTN:RETURN
                       :rem 254
360 PRINT:POKE198,0:INPUT#15,A
    $,B$,C$,D$         :rem 42
370 IFB$="OK"THENFL=0:RETURN
                       :rem 242
380 PRINTA$" "B$" "C$" "D$" ":
    FL=1:PRINT"{DOWN}HIT ANY K
    EY":A$=""          :rem 254
385 GETA$:IFA$=""THEN385
                       :rem 99
387 RETURN             :rem 130
390 REM====== FILE FOLLOWS ===
    ========           :rem 150
```

Page Flipping On The Atari

Clay Stuart

Page flipping is an animation technique in which entire screens can be flashed in rapid sequence, much like flipping through the pages of a book. This makes possible some amazing graphics displays. The article includes two demonstration programs (one for computers with as little as 16K RAM, and another for machines with at least 48K). The programs work on the 400/800, XL, and XE models.

Animation is any sequence of events that creates the illusion of motion. Note the phrase *illusion of motion*: no actual motion is required. For example, consider the lights on a movie marquee, stadium scoreboard, or message board. When the lights are flashed in sequence, they produce the illusion of motion, although the bulbs themselves are stationary.

Movies and cartoons work on the same principle. They consist of a series of still pictures, each slightly different than the one before it. When the pictures are projected in rapid succession, we perceive motion. The same principle can be applied to computers, except that few computers can draw high-resolution screens fast enough to fool the eye. The answer is a special programming technique known as *screen flipping* or *page flipping*. Screens are drawn beforehand and stored in memory, then displayed one after the other in an instant.

ANTIC And Display Lists

By now, many of you are well acquainted with the Atari ANTIC chip, display lists, and how Atari computers display screen images. Many articles and books have been published on this subject. However, a quick overview will be helpful here.

Besides the 6502 microprocessor, the Atari also contains a special chip known as ANTIC. ANTIC is a true microprocessor with its own instruction set, and it is in charge of displaying all screen data. The program that instructs ANTIC is called a *display list*. The display list is merely a list of instructions that locates screen memory for ANTIC and tells it which graphics mode to display. The starting address of the display list is stored in the customary low byte/high byte form at memory locations 560 and 561. To find the starting address of the display list in any graphics mode, type the following line in immediate mode:

```
GR.[mode]:DL=PEEK
(560)+256*PEEK(561):PRINT DL
```

where *[mode]* is the graphics mode number. This line can also be included in a program to store the display list address in the variable DL.

Screen Memory And Pointers

While display lists differ greatly, depending on the graphics mode, one thing is always certain. The fifth and sixth bytes of the display list contain the low and high bytes of the starting address of screen memory. To find the starting address of screen memory in any graphics mode, use this line:

```
GR.[mode]:DL=PEEK
(560)+256*PEEK(561):ST=PEEK
(DL+4)+256*PEEK(DL+5):PRINT ST
```

Remember that when indexing within a list, you start counting at zero. That's why the fifth element of the display list is calculated at DL+4 and the sixth at DL+5. These two bytes together are called a *pointer*. As ANTIC scans the display list, these two bytes point ANTIC to the start of the screen memory.

The use of pointers is called *indirection*. Indirection can be a very powerful tool, and thanks to the foresight of Atari's engineers, it makes page flipping possible on Atari computers. Some other computers reserve an area of memory that is dedicated entirely to the video display. To display a new screenful of data, the entire screen must be redrawn. This requires the high speeds of machine language, and at best is a complicated and time-consuming project. Through the use of pointers and indirection, the Atari avoids these complications.

By simply POKEing new values into the screen memory pointer in the display list, any area of memory can be instantly displayed on the screen. There is only one important rule to follow. To avoid garbled displays, screen memory should not cross a 4K memory boundary—that is, any address evenly divisible by 4096.

There is a way around this problem, but there's no room to explain it in detail here. However, you should note that it's impossible to flip pages in GRAPHICS 8 without taking this problem into account. Because GRAPHICS 8 uses about 8K of screen memory, the screen always crosses a 4K boundary. For now, it's probably best to experiment with page flipping in graphics modes 0 through 7, which use less than 4K of screen memory. With careful planning, screen memory need never cross a 4K boundary in these modes.

Setting Up Page Flipping

To implement page flipping, you must first calculate the starting address of the area of memory you wish to display. (You'll have to determine how much memory to set aside depending on the number of pages you want to flip, how much RAM is installed in your machine, etc.) Next, convert this number to its low-/high-byte representation using the following line:

```
AD=[address]:HI=INT(AD/256):
LO=AD-(256*HI)
```

where *[address]* is the starting memory address.

Then, choose a graphics mode and calculate the starting address of the display list. Finally, POKE the values LO and HI into the screen

memory pointers in the display list—LO byte first, HI byte second. Remember, the screen memory pointers are always the fifth and sixth bytes of the display list. Use the following short program as an example:

```
10 AD=40960
20 HI=INT(AD/256):LD=AD-(
   256*HI)
30 GRAPHICS 0
40 DL=PEEK(560)+256*PEEK(
   561)
50 POKE DL+4,LD:POKE DL+5
   ,HI
```

When executed, the screen should almost instantly change to display this new area of memory. This area of memory is part of the BASIC ROM, and should be filled with all sorts of interesting data. It will be displayed as characters, since GRAPHICS 0 is a character mode. By substituting different values for the variable AD in the program above, any area of memory can be displayed (as long as it doesn't cross a 4K boundary, remember). Of course, if the area of memory you choose to display is empty, the screen will be blank.

More Pointers

There are two other memory locations that are important to know when page flipping. These are locations 88 and 89, another pointer. They store a memory address in the usual low-byte/high-byte form, and point to the area of memory where all PRINTs, PLOTs, and DRAWTOs will be directed, except for information displayed in text windows. By POKEing new values into these locations, you can redirect all PRINTs, PLOTs, and DRAWTOs to any area of memory. In other words, rather than printing text or drawing graphics on the screen, you can print text or draw graphics anywhere in memory and then display this page instantly.

This makes it possible to construct a series of pictures, each slightly different than the one before it, and each in a separate area of RAM. By rapidly flipping through these pictures in sequence, a program creates the illusion of motion. That's how the demonstration programs below work.

A Spinning Globe

If you have at least 48K RAM, use Program 1. If you have only 16K RAM, use Program 2 and make sure no disk drive is connected. Program 1 creates a spinning globe on the high-resolution GRAPHICS 8 screen. Program 2 creates a spinning globe too, but on the medium-resolution GRAPHICS 7 screen to conserve memory. (Program 2 also works on machines with more than 16K as long as the disk drive is disconnected.)

Both programs GOSUB to a routine that READs machine language DATA into page 6 (starting at location 1536 decimal). After RETURN, Program 1 fills an array with SIN and COSINE values to speed up the drawing process. Program 2 skips this step due to the limited memory on a 16K machine. Both programs switch to a PLOT mode at line 60 and set up the various color registers at line 70. The starting address of the display list is calculated at line 80, and certain variables are initialized at line 90. (When you run Program 1, the screen is blank for about one minute during these steps.)

Lines 100 through 240 use BASIC trigonometric functions to draw three slightly different views of a wire globe. Each drawing is stored in a separate area of RAM. The spokes of this globe are nine degrees apart, and each drawing shows the globe rotated three degrees from the previous one. As each drawing is completed, the ON-GOSUB statement at line 230 directs the program to one of two subroutines at lines 1000 and 2000. These routines POKE the screen memory pointers into the display list, and POKE the PRINT, PLOT, and DRAWTO pointers with the address of the next area of memory to be displayed.

When all of this is completed, line 250 calls the machine language subroutine in page 6. This sets up a vertical blank interrupt routine that rapidly displays the drawings in sequence. The globe appears to spin!

Multiprocessing?

Although machine language is not really necessary for page flipping, it was included here for a reason. Notice that when the globe starts spinning, the READY prompt appears in the text window at the bottom of the screen. You can type LIST to view the program in the text window *while it is executing*, or even type NEW and enter another program without affecting the display. This allows you to incorporate the spinning globe in your own programs.

Another interesting item is memory location 1554. This location in the vertical blank routine controls the speed of the globe's rotation. It normally contains a 1. Try POKEing 1554 with a 3 to slow down the rotation, or a 0 to speed things up.

For an example of page flipping in BASIC, add these lines to Program 1:

```
LJ 245 PDKE DL+5,129:PDKE DL
       +101,144:FOR X=1 TO 1
       0:NEXT X
IJ 246 PDKE DL+5,97:POKE DL+
       101,112:FDR X=1 TD 10
       :NEXT X
FJ 247 PDKE DL+5,65:PDKE DL+
       101,80:FDR X=1 TD 10:
       NEXT X
HC 248 GDTD 245
```

or these lines to Program 2:

```
AJ 245 PDKE DL+5,16:FOR X=1
       TD 10:NEXT X
AI 246 PDKE DL+5,32:FOR X=1
       TD 10:NEXT X
BA 247 PDKE DL+5,48:FOR X=1
       TD 10:NEXT X
HC 248 GDTD 245
```

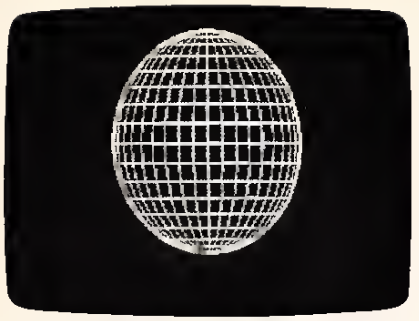
These modifications bypass the machine language, yet show how BASIC is plenty fast enough for page flipping. You can slow down or speed up the globe by changing the values in the FOR-NEXT loops. Or you can press BREAK and enter GOTO 250 in direct mode to let the machine language routine spin the globe.

Feel free to use these programs for your own pursuits. Remember that once they are running, you can enter NEW and type in your own program. Who knows—maybe you can come up with a game that has a rotating planet in the background. (Incidentally, player/missile graphics is an ideal way to add moving objects when flipping screens, because it's not affected by page flipping.)

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering these listings.

Program 1: Spinning Globe Demo (48K)

```
EO 5 REM REQUIRES AT LEAST 4
   8K
FB 10 POKE 559,0:GOSUB 3000:
   DIM ARR(450,1):DEG
BK 19 REM **** ARRAY FILL RO
   UTINE ****
DL 20 FDR I=0 TO 450
PP 30 ARR(I,0)=SIN(I)
PM 40 ARR(I,1)=CDS(I)
```

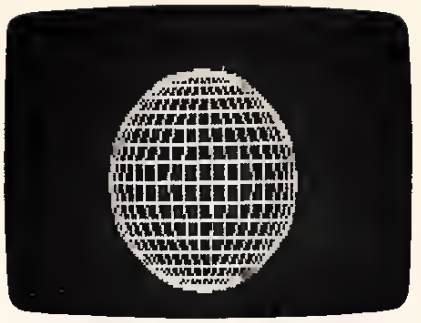


Quickly flipping in sequence through four previously drawn screens, Program 1 creates the illusion of a spinning globe in high-resolution graphics.

```
ON 50 NEXT I
AB 59 REM ** SETUP GRAPHICS
      MODE **
OP 60 GRAPHICS 8
KF 70 POKE 710,144:POKE 712,
      144:CDLOR 1
DN 79 REM ** CALCULATE START
       OF OL **
EK 80 OL=PEEK(560)+256*PEEK(
      561)
AO 90 X0=159:Y0=79:ROS=78:Z=
      1
GO 99 REM **** DRAW GLOBE RO
      UTINE ****
EP 100 FOR ROT=6 TO 0 STEP -
      3
FC 110 FOR ANG=ROT TO RDT+18
      0 STEP 9
JD 120 R=ROS*ARR(ANG,1)
IE 130 X=X0+R*ARR(270,1)
BN 140 Y=Y0+ROS*ARR(270,0)
LC 150 PLOT X,Y
AN 160 FOR CIR=280 TO 450 ST
      EP 10
BK 170 X=R*ARR(CIR,1)
LC 180 Y=RDS*ARR(CIR,0)
KP 190 DRAWTO X0+X,Y0+Y
CN 200 IF ANG=ROT THEN DRAWT
      O X0-X,Y0+Y:PLOT X0+X
      ,Y0+Y
LA 210 NEXT CIR
KJ 220 NEXT ANG
LL 230 ON Z GOSUB 1000,2000
NK 240 NEXT ROT
LJ 245 POKE DL+5,129:POKE DL
      +101,144:FOR X=1 TO 1
      0:NEXT X
IJ 246 POKE DL+5,97:POKE DL+
      101,112:FOR X=1 TO 10
      :NEXT X
FJ 247 POKE DL+5,65:POKE DL+
      101,80:FOR X=1 TO 10:
      NEXT X
HC 248 GOTO 245
LO 250 A=USR(PAGE6)
GP 260 END
ED 999 REM * POINTER POKE SU
      BROUTINE *
PC 1000 POKE OL+5,97:POKE OL
      +101,112
PO 1010 POKE 89,97
IN 1020 Z=2
KE 1030 RETURN
HE 1999 REM * POINTER POKE S
      UBROUTINE *
NC 2000 POKE OL+5,65:POKE OL
      +101,80
PK 2010 POKE 89,65
IL 2020 Z=0
KF 2030 RETURN
NG 2999 REM **** ML READ SUB
      ROUTINE ****
CC 3000 PAGE6=1536
HB 3010 FOR I=0 TO 58
LM 3020 READ O8J
MO 3030 POKE PAGE6+I,O8J
EP 3040 NEXT I
KI 3050 RETURN
LN 3999 REM ** MACHINE LANGU
      AGE DATA **
BI 4000 DATA 104,169,5,133,2
      04,169,6,162
LM 4010 DATA 6,160,15,32,92,
      228,96,165
DA 4020 DATA 20,41,1,208,29,
      164,204,185
PB 4030 DATA 53,6,141,85,128
      ,136,185,53
KP 4040 DATA 6,141,181,128,1
      92,0,240,6
LK 4050 DATA 136,132,204,76,
      50,6,169,5
CO 4060 DATA 133,204,76,95,2
      28,80,65,112
OC 4070 DATA 97,144,129
```


Program 2: Spinning Globe Demo (16K)

```
KI 5 REM DISCDNNECT DISK DRI
     VE
FN 10 PDKE 106,64:GRAPHICS 0
      :POKE 559,0:GDSUB 3000
DF 50 DEG
AB 59 REM ** SETUP GRAPHICS
      MODE **
OO 60 GRAPHICS 7
KF 70 POKE 710,144:POKE 712,
      144:CDLOR 1
DN 79 REM ** CALCULATE START
       DF DL **
```



Program 2 also simulates a spinning globe via page flipping, but uses medium-resolution graphics to save memory.

```
EK 80 OL=PEEK(560)+256*PEEK(
      561)
NH 90 X0=79:Y0=39:RDS=38:Z=1
GO 99 REM **** DRAW GLOBE RO
      UTINE ****
EP 100 FOR ROT=6 TO 0 STEP -
      3
FC 110 FOR ANG=RDT TO ROT+18
      0 STEP 9
EB 120 R=ROS*COS(ANG)
CH 130 X=X0+R*COS(270)
MG 140 Y=Y0+RDS*SIN(270)
LC 150 PLOT X,Y
AN 160 FOR CIR=280 TO 450 ST
      EP 10
LN 170 X=R*CDS(CIR)
FL 180 Y=RDS*SIN(CIR)
KP 190 DRAWTO X0+X,Y0+Y
CN 200 IF ANG=ROT THEN DRAWT
      O X0-X,Y0+Y:PLOT X0+X
      ,Y0+Y
LA 210 NEXT CIR
KJ 220 NEXT ANG
KK 230 ON Z GOSUB 1000,2000,
      3000
MK 240 NEXT ROT
LO 250 A=USR(PAGE6)
GP 260 END
ED 999 REM * POINTER POKE SU
      BROUTINE *
HB 1000 POKE OL+5,32
PO 1010 POKE 89,32
IN 1020 Z=2
KE 1030 RETURN
HE 1999 REM * POINTER POKE S
      UBROUTINE *
HE 2000 POKE DL+5,16
PG 2010 POKE 89,16
IL 2020 Z=0
KF 2030 RETURN
NG 2999 REM **** ML READ SUB
      ROUTINE ****
CC 3000 PAGE6=1536
GP 3010 FOR I=0 TO 47
LM 3020 READ O8J
MO 3030 POKE PAGE6+I,O8J
EP 3040 NEXT I
KI 3050 RETURN
LN 3999 REM ** MACHINE LANGU
      AGE DATA **
BF 4000 DATA 104,169,2,133,2
      04,169,6,162
LN 4010 DATA 6,160,15,32,92,
      228,96,165
MI 4020 DATA 20,41,1,208,21,
      164,204,185
LF 4030 DATA 45,6,141,167,47
      ,192,0,240
FO 4040 DATA 5,198,204,76,42
      ,6,169,2
PO 4050 DATA 133,204,76,95,2
      28,16,32,48
```

Commodore 64 Hi-Res Quick Clear

Paul W Downing

Here's a machine language routine that clears the Commodore 64's high-resolution screen in less than a second.

If you've ever used high-resolution graphics on the Commodore 64, you probably know how long it takes to clear the hi-res screen in BASIC. You need to POKE 8,000 memory locations with zeros, usually with a line like this:

```
FOR J=8192 TO 16192:POKE J,0:NEXT
```

This takes about 30 seconds—not a very long time, but it can seem almost endless if you're staring at a PLEASE WAIT message on the screen, waiting for the program to set up.

"Quick Clear" is a short machine language routine that clears the hi-res screen in less than one second. It can be inserted in any program that uses high-resolution graphics. You don't need to understand machine language to use it—just type it in and enter RUN. This installs the ML in the cassette buffer, starting at location 828. Once the routine is in place, use SYS 828 whenever you want to clear the hi-res screen.

Fill With Any Value

If you've never used hi-res graphics before, try this experiment. Run Quick Clear, then enter the following line to put your 64 in bitmap mode:

```
POKE 53265,PEEK(53265)OR32:
POKE 53272,PEEK(53272)OR8
```

The screen will be full of garbage. Now press SHIFT–CLR/HOME and enter SYS 828. The screen will clear in a heartbeat.

Ordinarily, you'll want to clear the screen with zeros. But you can also use this routine to fill the hi-res screen with any value from 0 to 255. Just change the second number in the third DATA statement from 0 to the desired number.

The ML routine is relocatable, so you don't need to put it in the cassette buffer. To change its location, change the variable SA in line 10 from 828 to another safe address (49152, for example).

The variable HS in line 10 is the starting address of the hi-res screen. If you locate the screen at some address other than 8192, be sure to change HS to match.

Quick Clear

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
10 SA=828:HS=8192:POKE2,HS-256
   *INT(HS/256):POKE3,INT(HS/2
   56)                    :rem 30
20 FORA=SATOSA+31:READB:POKEA,
   B:NEXT                 :rem 74
30 DATA 165,2,133,4,165,3
                          :rem 193
40 DATA 133,5,162,30,160,0
                          :rem 233
50 DATA 169,0,145,4,136,208
                          :rem 45
60 DATA 251,230,5,202,16,242
                          :rem 81
70 DATA 160,64,145,4,136,16
                          :rem 45
80 DATA 251,96          :rem 181
```

Unlocking IBM BASIC Programs

Peter F. Nicholson

This short utility unlocks BASIC programs which have been saved in protected format with the P option. It works on any IBM PC or PCjr.

IBM BASIC lets you save a program in three formats: in tokenized (compressed binary) form, as an ASCII file, or as a protected (encoded binary) program. The commands for these options are:

```
SAVE "filename"      (tokenized)
SAVE "filename",A    (ASCII)
SAVE "filename",P    (protected)
```

In each case, DOS automatically appends the extender .BAS and does not indicate the format on disk directories. You can load a program saved in any format with LOAD "*filename*.BAS", omitting the .BAS extender if you wish.

Although a protected program loads and runs normally, it cannot be listed or edited, and neither BASIC nor DOS provides a way to "unlock" it. So when you save a program in protected format, you should also save an unprotected copy in case you decide to make some changes later. If you find yourself without a backup, however, the following utility can remove the protection.

Type in and save UNPROT.UTL. Note that *you must save it with the filename UNPROT.UTL*. When you run it, you'll be prompted to enter the active drive (enter A if you have one drive) and the name of the protected program. The drive runs briefly as the protection is removed, and then your program is listed on the screen, ready for you to edit or resave.

Invisible Fingers

To mimic the effect of entering direct keyboard commands, UNPROT.UTL assigns strings to the ten special function keys. It then manipulates the keyboard buffer to enter each string automatically, as if the function keys were being pressed in sequence by invisible fingers. If you use DOS 2.1, the subroutine at line 2000 automatically enters the function keystrokes for you. If you have another version of DOS, you'll have to delete the GOSUB 2000 statement from line 290 and press F1 through F10 in sequence yourself, after entering the filename.

Mimicking keystrokes is an efficient technique, but it makes a program somewhat difficult to follow. If you're interested in how this utility works, here's a brief explanation of how protected programs can be unlocked.

The Key Addresses

The first thing you need to learn is where the program starts and ends in memory. As explained in Appendix I of the *IBM BASIC Manual*, these addresses can be found with the following PEEKs:

```
PEEK(&H30)+256*PEEK(&H31)
                Program starting address
PEEK(&H358)+256*PEEK(&H359)-1
                Program ending address
```

The starting address is the same in every case; you can find it simply by entering NEW followed by the first PEEK statement above. Finding the ending address is more difficult, as you'll find if you load a protected program and enter the second statement. All you'll get for your trouble is an illegal function call error.

However, there's another way to get the same information. Scalar variables are stored immediately after the end of a BASIC program, and the VARPTR function can find the address of any variable. All you need to do is define an arbitrary scalar variable, CHAIN the protected program into memory, and use VARPTR to find the address of the dummy variable.

Breaking The Chains

Unlike the LOAD command, which clears variables, CHAIN brings a program into memory and begins execution at a specified line number without destroying preexisting variables. This is the method used in UNPROT.UTL. We don't want to *run* the chained program after it's in memory, so the CHAIN command uses a nonexistent line number (65529). This simply halts execution with an illegal function call error.

Subtracting a few bytes to account for the variable descriptor gives us the exact address where the program ends. To determine its length, we subtract the starting address from the ending address.

Now that we know the program's starting address and length, we BSAVE it back to disk as a binary file. After performing a second NEW, it's necessary to set the pointers for the start of scalar variables, arrays, and strings at the spot where the program ends. Finally, the program is BLOADed back into memory at the correct starting address, and the unlocking process is complete.

If you would rather not type in this program, send a formatted disk with a self-addressed, postage-paid mailer and a $3.00 check to:

Peter F. Nicholson
1701 South Princeton Road
Ottawa, Kansas 66067

UNPROT.UTL

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing

```
ID BØ REM 'UNLOCK' PROGRAMS SAVE
      D IN PROTECTED FORMAT.
MA 9Ø REM LOADS A PREVIOUSLY PRO
      TECTED PROGRAM INTO MEMORY
MB 95 REM WITHOUT PROTECTION, SO
       PROGRAM CAN BE LISTED AND
       SAVED.
KD 96 REM IBM BASIC VERSIONS 1.1
      AND 2.Ø
JL 1ØØ DEF SEG:CLEAR:KEY OFF:CLS
      :ON ERROR GOTO 3ØØ
HJ 11Ø B%=Ø:A=Ø
NM 12Ø GOSUB 1ØØØ
```

```
KD 130 A=PEEK(&H30)+256*PEEK(&H3
       1)
IL 140 LINE INPUT "PROTECTED FIL
       E DRIVE ";G$:IF LEN(G$)>0
        THEN IF INSTR(G$,":")=0
       THEN G$=G$+":"
HG 150 LINE INPUT "PROTECTED FIL
       E NAME ";F$:IF INSTR(F$,"
       .")=0 THEN F$=F$+".BAS"
HP 160 G$=G$+F$
DI 170 F$="PROT.SCR"
HG 180 H$="PROT.DAT":I$="UNPROT.
       UTL"
HK 190 KEY 1,"B=VARPTR(B)"+CHR$(
       13)
IF 200 KEY 2,"BSAVE F$,A,B-A"
DL 210 KEY 3,"-4:BSAVE H$,B,"
ML 220 KEY 4,"4:CHAIN I$,500"+CH
       R$(13)
DM 230 KEY 5,"BLOAD"+CHR$(34)
ME 240 KEY 6,H$+CHR$(34)+",856"+
       CHR$(13)
EI 250 KEY 7,"BLOAD"+CHR$(34)+F$
IH 260 KEY 8,CHR$(34)+","+STR$(A
       )+CHR$(13)+"LIST"+CHR$(13
       )
AC 270 KEY 9,"FOR I=1TO 10:KE"
HN 280 KEY 10,"Y I,"+CHR$(34)+CH
       R$(34)+":NEXT"+CHR$(13)
LJ 290 GOSUB 2000:COLOR 0,0:CHAI
       N G$,65529',ALL
EE 300 FOR I=1 TO 10:KEY I,"":NE
       XT I
DH 310 COLOR 7,0:IF ERL=290 AND
       ERR=53 THEN CLS:BEEP:PRIN
       T G$+" DOES NOT EXIST":RE
       SUME 140
ME 320 ON ERROR GOTO 0:END
BB 500 B=0:DIM B1%(2):COLOR 7,0
DD 510 BLOAD "PROT.DAT",VARPTR(B
       )
DB 520 FOR I=0 TO 2:IF B<2^15 TH
       EN B1%(I)=B ELSE B1%(I)=B
       -2^16
MC 530 NEXT I:BSAVE "PROT.DAT",V
       ARPTR(B1%(0)),6
DC 540 NEW
HH 1000 PRINT "UNPROTECTING BASI
       C PROGRAMS"
GJ 1010 LOCATE 4,10:PRINT "1. YO
       U WILL BE PROMPTED FOR T
       HE FILE DRIVE AND NAME"
PC 1040 LOCATE 7,10:PRINT "2. TH
       E FINAL STEP IS THE LIST
       ING OF YOUR PROGRAM"
GF 1050 LOCATE 16,1:PRINT "NOTE:
        FUNCTION KEYS ARE CLEAR
       ED BY THE PROGRAM AND TW
       O SCRATCH"
KH 1060 LOCATE 18,1:PRINT "FILES
       , PROT.SCR AND PROT.DAT
       ARE LEFT ON YOUR DEFAULT
        DRIVE"
DH 1070 LOCATE 25,1:PRINT "PRESS
        ANY KEY TO START";
CB 10B0 KB$=INKEY$:IF KB$="" GOT
       O 1090: 'CLEAR KEYBOARD
EH 1090 KB$=INKEY$:IF KB$="" GOT
       O 1090
EF 1100 CLS:RETURN
BF 2000 REM ** SET KEYBOARD BUFF
       ER TO ENTER F1 THROUGH F
       10 AUTOMATICALLY***
DM 2010 REM **                  IBM P
       C DOS VERSION 2.1
                               ***
JB 2020 DEF SEG=&H40:FOR I=1 TO
       10:POKE 2*I+28,0:POKE 2*
       I+29,58+I:NEXT I
CO 2030 POKE 2*I+28,13:POKE 2*I+
       29,28:POKE 26,30:POKE 28
       ,50:DEF SEG:RETURN
```

Fast Atari Circles

Owen Sexsmith

Draw circles, stars, diamonds, and other geometric shapes in Atari BASIC quickly and easily, all with a single subroutine written in machine language. For the Atari 400/800, XL, and XE computers.

Unlike some newer versions of the language, Atari BASIC has no CIRCLE statement, so drawing circles can be a slow and cumbersome process. But with "Fast Atari Circles," you can easily draw circles, ellipses, stars, and a galaxy of other shapes. The routine is written in machine language for maximum speed, but you don't need to understand ML to use the routine in your own programs.

Type in and save Fast Atari Circles. When you run it, you'll see a pattern of finely drawn, elliptical lines. After that, the program generates colored disks, open stars, hexagons, diamonds, and other complex shapes in various graphics modes.

Believe it or not, all these shapes were created with just one routine. As you can see from the program, GOSUB 900 is used whenever graphics are generated. Line 900 consists of a single USR statement (which calls the machine language subroutine), followed by RETURN. The USR statement includes several descriptively named variables, such as XCENTER and YRADIUS. To create a shape, you simply assign values to these BASIC variables, then call the Fast Circles routine with USR.

Defining The Variables

XCENTER and YCENTER locate your shape on the screen. XCENTER defines the X coordinate, or horizontal location. Give XCENTER a small value to put the shape near the left of the screen, and larger values to move it to the right. YCENTER defines the Y coordinate, or vertical location. Smaller YCENTER values put the shape higher on the screen, and larger values move it down. XRADIUS and YRADIUS define the shape's width and height, respectively.

To learn how these four variables interact, look at lines 165–220. In line 165, XCENTER and YCENTER are given values that place the shape in the middle of the screen. In lines 205–220, a FOR-NEXT loop increases the values of XRADIUS and YRADIUS each time the loop is executed. Since XRADIUS is always greater than YRADIUS, the shape is wider than it is high, forming an ellipse. In lines 225–240, the process is repeated, giving YRADIUS greater values than XRADIUS, so the ellipse is higher than it is wide.

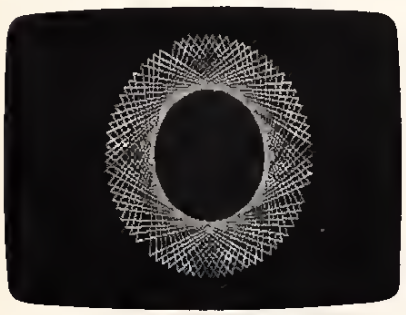
By changing the STEP values in lines 205 and 225, you can change the distance between the lines. For example, try STEP 3 instead of STEP 5 in line 205, and STEP 1 instead of STEP 5 in line 225. You'll see an interesting moiré effect in areas where the two shapes overlap. If you'd like to experiment further, add these two lines:

```
201 XRADIUS=96: YRADIUS=96:
    GOSUB 900
202 GOTO 202
```

When you run the program again, it draws one shape and pauses in an endless loop at line 202. Since XRADIUS and YRADIUS are equal, the shape is a circle (some TV sets and monitors may be mildly distorted, making the circles look slightly elliptical).

Press BREAK to stop, and edit line 175 so that DELTA=32. When the program runs, you should see an octagon. When DELTA=64, it becomes a diamond. If you change DELTA to an odd value such as 81,

the program draws a complex series of lines that eventually overlap to form a thick doughnut shape. When you're done experimenting with this section, delete lines 201 and 202.



A small example of some shapes you can draw with "Fast Atari Circles."

Using The Routine

The demonstration program contains REMarks explaining what each section does. By studying the program and experimenting with other sections, you can quickly learn how to handle all the variables used by the Atari Fast Circles routine.

To use this routine in your own programs, you'll need to include lines 50–60, 70, 900, and 1000–1135. Lines 50–60 create a table of sine values in SINE$. Line 70 builds the machine language routine in CIRC$. Line 900 contains the line-drawing USR call, and lines 1000–1135 are the machine language data. Put the lines that create CIRC$ and SINE$ in the setup portion of your program. Once the setup is complete, you're ready to create your own graphics masterpieces.

Fast Atari Circles

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
PL 10  GRAPHICS 2+16:POSITION
       5,4:? #6;"PLEASE WAIT
       ";? #6:? #6;"
       {7 SPACES}loading"
EI 40  REM
HM 42  REM BUILD A SINE TABLE
        IN SINE$
EN 45  REM
HM 50  DEG :DIM SINE$(65):FOR
        I=0 TO 64:X=INT(256*S
       IN(90/64*I)+0.5)
IL 55  IF X>255 THEN X=255
FI 60  SINE$(I+1)=CHR$(X):NEX
       T I
EP 65  REM
EA 66  REM PUT M.L. ROUTINE I
       N CIRC$
FB 67  REM
PB 70  DIM CIRC$(280):FOR I=1
        TO 280:READ X:CIRC$(I
       )=CHR$(X):NEXT I
HK 150 REM
DA 155 REM ELLIPSES
HL 160 REM
IJ 165 XCENTER=160:YCENTER=9
       6
LN 170 ARCSTART=0:ARCEND=0
KF 175 KOLOR=1:DELTA=2
KN 200 GRAPHICS 8+16
LE 205 FOR I=0 TO 75 STEP 5
DG 210 XRADIUS=5+2*I:YRADIUS
       =5+I
LB 215 GOSUB 900
BN 220 NEXT I
LD 225 FOR I=0 TO 45 STEP 5
DI 230 XRADIUS=5+I:YRADIUS=5
       +2*I
LD 235 GOSUB 900
BO 240 NEXT I
DB 245 FOR I=1 TO 300:NEXT I
HL 250 REM
PO 255 REM DISKS
HM 260 REM
PI 265 KOLOR=1:DELTA=1:GRAPH
       ICS 7+16
LH 270 FOR I=6 TO 27 STEP 3
DJ 275 XCENTER=10+4*I:YCENTE
       R=10+2*I
CE 280 FOR J=0 TO I
IH 285 XRADIUS=J:YRADIUS=J:A
       RCEND= NOT J
LE 290 GOSUB 900
CJ 295 NEXT J
BP 300 KOLOR=KOLOR+1:IF KOLO
       R>3 THEN KOLOR=1
CA 305 NEXT I
HI 310 REM
KB 315 REM SQUARES
HJ 320 REM
PL 325 GRAPHICS 3+16:DELTA=6
       4
FL 330 REM SIXTY-FOUR IS A Q
       UARTER ARC
EH 335 XCENTER=20:YCENTER=12
MD 337 FOR K=0 TO 1:ARCSTART
       =32*K:ARCEND=32*K
NC 340 FOR I=0 TO 5:KOLOR=KO
       LOR+1:IF KOLOR>3 THEN
        KOLOR=1
AD 345 FOR J=0 TO 1
MB 350 XRADIUS=2*I+J:YRADIUS
       =2*I+J
LC 360 GOSUB 900
KJ 370 NEXT J:NEXT I:NEXT K
DF 375 FOR I=1 TO 300:NEXT I
HP 380 REM
PO 385 REM VARIOUS OTHER SHA
       PES
IA 390 REM
LI 400 GRAPHICS 7+16:KOLOR=0
       :DIM S(4),E(4),I(4)
IP 405 S(1)=0:E(1)=0:I(1)=64
       :S(2)=193:E(2)=191:I(
       2)=102
MG 410 S(3)=16:E(3)=16:I(3)=
       32:S(4)=0:E(4)=0:I(4)
       =1
EN 420 XRADIUS=10:YRADIUS=10
CM 430 FOR I=0 TO 3:S=4-I:FO
       R J=0 TO 3
CE 440 KOLOR=KOLOR+1:IF KOLO
       R>3 THEN KOLOR=1
FG 450 S=S+1:IF S>4 THEN S=1
EJ 460 XCENTER=20+40*J:YCENT
       ER=12+20*I
AD 470 ARCSTART=S(S):ARCEND=
       E(S):DELTA=I(S)
LF 480 GOSUB 900
DI 490 NEXT J:NEXT I
DI 495 FOR I=1 TO 300:NEXT I
GD 500 GOTO 500
ED 900 X=USR(ADR(CIRC$),ADR(
       SINE$),XCENTER,YCENTE
       R,XRADIUS,YRADIUS,ARC
       START,ARCEND,256*KOLO
       R+DELTA):RETURN
EF 1000 DATA 104,104,133,231
        ,104,133,230,104,133
        ,217
FJ 1005 DATA 104,133,216,104
        ,133,228,133,229,104
        ,133
EK 1010 DATA 218,104,104,133
        ,219,104,104,133,220
        ,104
EI 1015 DATA 104,133,221,104
        ,104,133,222,104,141
        ,251
JB 1020 DATA 2,104,133,223,1
        62,0,134,227,165,227
DB 1025 DATA 56,233,64,133,2
        27,165,221,56,229,22
        7
CP 1030 DATA 133,224,144,240
        ,165,227,41,128,133,
        225
FE 1035 DATA 165,227,41,64,2
        40,7,169,64,56,229
PK 1040 DATA 224,133,224,165
        ,227,240,6,201,192,2
        40
AG 1045 DATA 2,162,128,134,2
        26,164,224,177,230,1
        33
PH 1050 DATA 214,165,220,133
        ,215,169,0,133,212,1
        62
CN 1055 DATA 8,70,214,144,3,
        24,101,215,106,102
NA 1060 DATA 212,202,208,243
        ,166,225,240,5,73,25
        5
DP 1065 DATA 24,105,1,24,101
        ,218,133,84,169,64
EH 1070 DATA 56,229,224,168,
        177,230,133,214,165,
        219
DL 1075 DATA 133,215,169,0,1
        33,212,162,8,208,2
JG 1080 DATA 208,138,70,214,
        144,3,24,101,215,106
GC 1085 DATA 102,212,202,208
        ,243,133,227,166,226
        ,240
DJ 1090 DATA 16,165,216,56,2
        29,227,133,85,165,21
        7
EA 1095 DATA 233,0,133,86,24
        ,144,13,165,216,24
JK 1100 DATA 101,227,133,85,
        165,217,105,0,133,86
DC 1105 DATA 166,228,208,40,
        162,96,134,228,169,1
        1
LF 1110 DATA 157,66,3,169,0,
        157,72,3,157,73
BK 1115 DATA 3,173,251,2,32,
        86,228,169,17,157
LE 1120 DATA 66,3,169,12,157
        ,74,3,169,0,157
PB 1125 DATA 75,3,240,9,162,
        96,32,86,228,166
PF 1130 DATA 229,208,16,165,
        221,24,101,223,133,2
        21
EF 1135 DATA 197,222,208,142
        ,202,134,229,208,137
        ,96
```

Apple Universal INPUT

William Simpson

Banish EXTRA IGNORED errors from your Applesoft programs with this short INPUT routine. It works on any Apple II series computer with DOS 3.3 or ProDOS.

As you know if you've ever tried it, Applesoft BASIC won't let you type commas or colons when responding to an INPUT prompt. The computer rejects everything after the punctuation and gives you an EXTRA IGNORED error. There's a good reason for this, but there may be times when you'd like an input string to include the punctuation. For example, you might want to input a time value in response to a prompt like:
ENTER HOURS:MINUTES.

"Apple Universal Input" solves this problem and can be used as a routine in any Applesoft BASIC program. Once installed, it lets you input strings containing commas and colons, from the keyboard or from disk.

Type in and save the following program, then enter RUN and type any string containing commas or colons. The program prints the string to show that the input was accepted without errors.

You'll notice that the input prompt is a greater-than sign (>) rather than a question mark. This signals that the normal Applesoft INPUT command is not in use. If you don't like this prompt, you can easily change it to another character. Find the ASCII code for the character you prefer, add 128 to the ASCII code, and substitute that value for the second DATA number in line 270 of the program. For example, the < character has an ASCII code of 60. To use that character as the prompt, you would replace the second DATA number in line 270 with 188 (60 + 128).

Program Breakdown

Let's look at the example program to learn how this input routine can be used in other programs.

Line 100 defines the variable T$. It's essential that this be the first variable your program defines.

Line 110 POKEs a short machine language (ML) routine into memory; the DATA for this routine is contained in lines 270–300. Lines 120 and 130 print a prompt on the screen, and call the new input routine with GOSUB 190. When using this routine in your own programs, you should use a similar GOSUB whenever you want to input a new string. Note that the string is returned in the variable A$ (line 140).

The BASIC subroutine calls the ML routine (CALL 768) to bring the input string into the computer's memory. Using the ROM GETLN routine, the ML routine first moves the string into the input buffer. Then it stores the string's length in location 798, subtracts 128 from each character's value to obtain the correct ASCII codes, and returns control to BASIC.

Lines 200–260 move the string from the input buffer to a safe place in memory where it can accessed by the main program. The vehicle for this transfer is the string variable T$, which you'll recall was the first variable defined in the program. This is done so that you can find the descriptor for T$ by PEEKing the pointer in locations 105–106.

Variable Descriptors

As you may know, a simple variable descriptor consists of five bytes in the following form:

Byte #	Function
1 =	First letter of the variable's name
2 =	Second letter of the name
3 =	Length of the variable
4 =	Low byte of the variable's memory address
5 =	High byte of the variable's memory address

By manipulating the descriptor for the variable T$, it is relatively simple to transfer the string from the input buffer (where it would quickly be overwritten) to another string variable (A$ in example program).

After the descriptor is located (line 210), its third byte is POKEd with the length of the string (line 220), and the fourth and fifth bytes are POKEd with the low byte/high byte address of the input buffer (lines 230–240). T$ is now set to the correct length and its descriptor points to the input buffer.

The final step (line 250) is to copy T$ into A$, using a form of the MID$ function that extracts every character from T$. You may substitute other names for T$ and A$, of course, when using this routine in your own programs.

Applesoft Universal Input

```
100 T$ = ""
110 FOR I = 768 TO 798: READ A:
    POKE I,A: NEXT
120 HOME
130 PRINT "INPUT ANYTHING": GOS
    UB 190
140 PRINT "ANYTHING==> ";A$
150 PRINT
160 INPUT "ANY MORE? (Y OR N) "
    ;YT$
170 IF YT$ = "Y" THEN 120
180 END
190 CALL 768
200 81 = PEEK (798)
210 82 = PEEK (106) * 256 + PEE
    K (105)
220 POKE 82 + 2,81
230 POKE 82 + 3,0
240 POKE 82 + 4,2
250 A$ = MID$ (T$,1)
260 RETURN
270 DATA  169,190,133,51,32,106
    ,253,142
280 DATA  30,3,164,0,204,30,3,2
    40
290 DATA  12,185,0,2,41,127,153
    ,0
300 DATA  2,200,76,12,3,96,0,0
```

Hardcopy Sprites For Commodore 64

Thomas H White

Make a hardcopy printout of any Commodore 64 sprite with this BASIC program for the VIC-1525, MPS-801, or MPS-802 printer. It's easily adapted to other printers as well.

Have you ever wished you could print a sprite on paper? For example, you may want to check some detail on a sprite you designed without bothering to load and run the program. With "Hardcopy Sprites," you can check such details in seconds rather than minutes, and even compile a personal library of sprite shapes for future reference.

Ordinary screen dump programs can't make sprite printouts because the data that defines sprite shapes isn't located in regular screen memory. To print out a sprite, you need a program that reads the 64-byte memory area where the sprite shape is actually stored.

Hardcopy Sprites uses this technique to print a 24 × 21 character representation of any sprite, with vertical or horizontal expansion if you wish. Large ball-shaped characters form single-color sprites, and additional characters represent multicolor sprites. Figures 1 and 2 show two examples of what is possible.

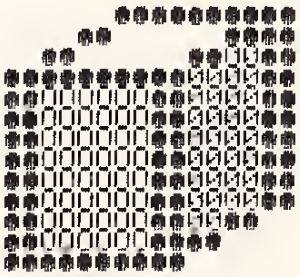
After typing and saving the program, POKE your sprite data into memory and note the address where it begins. Then simply run Hardcopy Sprites and follow the prompts. The program asks you to enter the memory address where the sprite data begins, to give the sprite a descriptive name, to choose horizontal or vertical expansion (or both), and to select single-color or multicolor mode.

Non-Commodore Printers

Hardcopy Sprites is written for Commodore printers, but is readily adapted to other printers. If your printer can't handle Commodore graphics, replace the ball-shaped Commodore graphics character (SHIFT-Q in the program listing) in lines 200, 210, 230, 340, and 360 with some other character, such as an X or a plus sign. With a little extra work, you could also incorporate Hardcopy Sprites as a subroutine within a BASIC sprite editor program.

Figure 1: Expanded Single-Color Sprite

Figure 2: Unexpanded Multicolor Sprite



Hardcopy Sprites

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing

```
10 REM ** SPRITE HARDCOPY **
                          :rem 30
20 INPUT"{CLR}{DOWN}SPRITE DAT
   A MEMORY LOCATION";PG
                          :rem 207
30 INPUT"{CLR}{DOWN}EXPAND X D
   IR. (Y/N)";EX$         :rem 35
40 INPUT"{7 SPACES}Y DIR. (Y/N
   )";EY$                 :rem 194
45 INPUT"{DOWN}SPRITE NAME";N$
                          :rem 243
47 INPUT"{DOWN}MULTICOLOR MODE
    (Y/N)";M$             :rem 82
50 OPEN1,4                :rem 40
55 PRINT#1,CHR$(15)"MEMORY LOC
   ATION"PG"{2 SPACES}"N$
                          :rem 242
60 FORR=0TO60STEP3        :rem 132
70 FORC=0TO2:P=PEEK(PG+R+C)
                          :rem 146
80 FORB=7TO0STEP-1:V(B)=P/2:P=
   INT(V(B))              :rem 111
90 V(B)=V(B)-INT(V(B)):rem 202
100 IFV(B)>0THENGOSUB200
                          :rem 184
110 IFV(B)=0THENGOSUB250
                          :rem 189
120 NEXTB                 :rem 20
130 FORB=0TO7:L$=L$+D$(B)
                          :rem 129
140 NEXTB:NEXTC           :rem 210
150 IFEY$<>"Y"THENPRINT#1,CHR$
    (15)+CHR$(16)+"28"+L$+CHR$
    (8)                   :rem 255
160 IFEY$="Y"THENFORRE=0TO1:PR
    INT#1,CHR$(15)+CHR$(16)+"1
    6"+L$+CHR$(8):NEXTRE
                          :rem 201
170 L$="":NEXTR           :rem 84
180 PRINT#1:CLOSE1:END:rem 108
200 D$(B)="Q"             :rem 223
210 IFM$="N"ANDEX$="Y"THEND$(B
    )="QQ"                :rem 29
220 IFM$="Y"ANDEX$="N"THENGOSU
    B300                  :rem 19
230 IFM$="Y"ANDEX$="Y"THEND$(B
    )="QQ":GOSUB320       :rem 121
240 RETURN                :rem 118
250 D$(B)=" "             :rem 19
260 IFM$="N"ANDEX$="Y"THEND$(B
    )="{2 SPACES}"        :rem 128
270 IFM$="Y"ANDEX$="N"THENGOSU
    B340                  :rem 28
280 IFM$="Y"ANDEX$="Y"THEND$(B
    )="{2 SPACES}":GOSUB360
                          :rem 224
290 RETURN                :rem 123
300 IFB=6ORB=4ORB=2ORB=0THENIF
    D$(B+1)=" "THEND$(B+1)="0"
    :D$(B)="0"            :rem 128
310 RETURN                :rem 116
320 IFB=6ORB=4ORB=2ORB=0THENIF
    D$(B+1)="{2 SPACES}"THEND$
    (B+1)="00":D$(B)="00"
                          :rem 226
330 RETURN                :rem 118
340 IFB=6ORB=4ORB=2ORB=0THENIF
    D$(B+1)="Q"THEND$(B+1)="O"
    :D$(B)="O"            :rem 147
350 RETURN                :rem 120
360 IFB=6ORB=4ORB=2ORB=0THENIF
    D$(B+1)="QQ"THEND$(B+1)="O
    O":D$(B)="OO"          :rem 4
370 RETURN          :rem 122
```

IBM Variable Lister

Peter F. Nicholson

This handy utility lists all the variables in your IBM BASIC programs. It runs on any PC with at least 64K RAM or PCjr with at least 128K RAM.

The ability to list variables is a valuable aid in debugging and documenting BASIC programs. The three programs accompanying this article show how IBM BASIC variables are stored and let you list the variables in any IBM BASIC program.

"Variable Lister" (Program 3) is the actual utility. Programs 1 and 2 demonstrate how IBM BASIC stores variables for those who are interested in the details (see below). But you don't have to be familiar with the theory to use Variable Lister.

The first step is to type in and save Program 3. (Note: All three programs assume that your screen is in the 80-column mode. Enter WIDTH 80 from BASIC or MODE 80 from DOS before running the programs.)

Once Variable Lister is saved on disk, there are two ways to run it. The easiest way is to run your own program first, break out, then activate Variable Lister with the CHAIN command. For example, if you have saved Variable Lister under the filename "VARLIST," enter CHAIN "VARLIST",70,ALL and press ENTER. This preserves your program's variables while Variable Lister loads and runs.

After doing this, Variable Lister replaces your program in memory. If you want to get a variable list while your program remains in memory, you need to renumber Variable Lister with line numbers not used in your program, and then merge it with your program using CHAIN MERGE. See the *IBM BASIC Manual* for details.

Of course, Variable Lister uses variables of its own. To avoid confusion, these variables (BAS, XLEN, CTA, CTV, II1, JJ1, AD0, VARNAME$, ARRNAME$, ITV!, and TESTARRAY) are suppressed from the variable list, and should not be used by your program.

The subroutine beginning at line 820 sorts variables and arrays alphabetically. There may be times when you'd rather display them chronologically (the order in which they're defined as the program runs). This is easily done by deleting the statement GOSUB 820 from lines 500 and 520.

How It Works

IBM BASIC uses four types of variables—integer, string, single-precision, or double-precision. The term *scalar* describes all variables that are not arrays. Appendix I of the *IBM BASIC Manual* explains how and where scalars are stored. Page I-2 of this appendix shows where the scalar storage area starts, and pages I-3 and I-4 explain the meaning of the data stored there.

If you want to see a graphic illustration of scalar variable storage, type in and save Program 1, then enter RUN. The program defines four variables—each a different type—and displays the *descriptor* bytes that describe each. You'll see two columns of numbers for each variable. The left column provides a reference number for each byte of the descriptor, and the right column shows the value that each byte contains.

The first byte of the descriptor indicates the variable type. This byte contains a 2 for an integer variable, 3 for a string, and 4 or 8 for a single- or double-precision variable, respectively. The next two bytes hold the first two characters of the variable name. For the variable A, we see the value of 65, the ASCII code for the letter A. The letter B is shown with an ASCII code of 66, and so on.

When a variable name has more than two characters, the fourth byte of the descriptor shows the number of remaining characters. The additional characters are stored immediately after this byte, with 128 added to their ASCII codes. Thus, where the variable is named ABC, you will see the values 65 and 66 (ASCII codes for AB), 1 (the number of remaining characters), and 195 (128 + ASCII code for C). The final bytes in each descriptor, highlighted in reverse video, show the value given to each variable.

Array Variables

The *IBM BASIC Manual* gives few details about how and where array variables are stored. However, its memory map does show that scalars, arrays, and strings occupy three contiguous areas in memory. You can use these formulas to find the boundaries of each zone:

PEEK(&H358)+256*PEEK(&H359)
Scalar variable space starts
PEEK(&H35A)+256*PEEK(&H35B)-1
Scalar variable space ends

PEEK(&H35A)+256*PEEK(&H35B)
Array space starts
PEEK(&H35C)+256*PEEK(&H35D)-1
Array space ends

PEEK(&H35C)+256*PEEK(&H35D)
String space starts

(Of course, the PEEKs won't return meaningful results until your variables have actually been declared. Prior to that time, the storage zones are empty, and the pointers all point to the same place.)

Like scalars, array variables can be any of four types: integer, string, single-precision, or double-precision. Thus, the first part of an array descriptor is the same as a scalar descriptor—first comes a type byte, followed by the characters of the array name—but additional bytes are needed to describe the more complex structure of an array. The two bytes after the name show the number of bytes needed to complete the array; this includes a description of the array's elements and dimensions, and the array data itself.

The next byte in the array descriptor (highlighted in reverse video) shows the number of dimensions in the array. For example, the statement DIM A(2,200) creates a two-dimensional array; the statement DIM B(1,2,3,4,5,6) creates an array with six dimensions. Although it's rarely necessary to use more than a few dimensions, IBM BASIC lets you define an array with as many as 255 of them.

The DIM statement that defines an array also specifies the maximum number of elements in each dimension. The array descriptor contains two more bytes for each dimension in the array, indicating the number of elements in that dimension. An array dimension may contain a maximum of 32,767 elements. If the number of elements is not specified, the default value of 10 elements is assigned.

Program 2 graphically illustrates array storage. Its display is similar to that of Program 1, using reverse video to highlight the area allocated for the array dimension and element numbers.

Lister Details

Strings stored in program lines may not be stored in string space. For example, say that your program has this line:

```
5 A$ = "A": B$ =
STRING$(2,CHR$(65))
```

Variable Lister reports both A$ and B$ as *variables*, but the string space is only two bytes long, since the character A for A$ is stored in line 5 of the program where it is defined.

Functions defined in a program are indicated by the FN prefix and are listed last for the variable type. A function is shown by adding 128 to the code of the first character. If your program contains DEF FNAA%, Variable Lister displays 2, 193 (65+128), 65, 0. Where your program consists of the line 5 A% = 1:DEF FNA%(X,Y)=X^2+Y^2, Variable Lister reports A, FNA, X, and Y as variables.

The program lists array dimensions exactly as defined in your DIM statement, independent of the OPTION BASE you have selected. If your program consists of the line 5 DIM A(2,2), the report should list A(2,2) and show 47 bytes occupied. If the program consists of 5 OPTION BASE 1: DIM A(2,2), the report should list A(2,2) and show 27 bytes occupied.

Keep in mind that string space is filled from the top of memory downward. To find the starting address of your stack area, use PEEK(&H30A) + 256 * PEEK (&H30B). Use PEEK(&H32F) + 256 * PEEK(&H330) to find the current string space. To get an accurate report, you may first need to issue a FRE("") command as described in the manual.

Finally, Variable Lister cannot report any variable that your program does not actually use. Consider this example:

```
5 D$="12-31-84":IF
RIGHT$(D$,2)="85" THEN A=1
```

Since the IF condition can never be satisfied, the computer never executes A=1. Variable Lister reports only one variable—D$.

If you would like a copy of this program, send a formatted disk with a self-addressed, postage-paid mailer and a check for $3.00 to:

Peter F. Nicholson
1701 South Princeton Road L
Ottawa, Kansas 66067

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering these listings.

Program 1: Scalar Variable Storage

```
GH 100 KEY OFF:DEF SEG:COLOR 7,0
       :CLS:PRINT "HOW SCALAR VA
       RIABLES ARE STORED":PRINT
LG 110 A%=0:AB$="0":ABC=0:ABCD#=
       0
ID 120 AD!=PEEK(&H35B)+256*PEEK(
       &H359)
PJ 130 PRINT "INTEGER        STRI
       NG        SINGLE PREC.
        DDUBLE PREC."
HH 140 FOR J=1 TO 4
GH 150 FOR I=0 TO PEEK(AD!)+PEEK
       (AD!+3)+3
QK 160 IF I>PEEK(AD!+3)+3 THEN C
       OLOR 0,7 ELSE COLOR 7,0
CL 170 K=PEEK(AD!+I)
FQ 180 LOCATE I+5,15*(J-1)+1:PRI
       NT USING "###    ";I;K;
QJ 190 NEXT I
MB 200 AD!=AD!+PEEK(AD!)+PEEK(AD
       !+3)+4
HE 210 NEXT J
GK 220 COLOR 7,0
FK 230 PRINT:PRINT "INTEGER  A%=
       0"
LQ 240 PRINT "STRING  AB$='0'"
CG 250 PRINT "SINGLE PRECISION
       ABC!=0"
JB 260 PRINT "DDUBLE PRECISION
       ABCD#=0"
CL 270 PRINT STRING$(5,CHR$(219)
       );" DATA ";
DP 280 LOCATE 15,1
```

Program 2: Array Variable Storage

```
EB 1000 KEY OFF:DEF SEG:COLOR 7,
        0:CLS:PRINT "HOW ARRAYS
        ARE STORED":PRINT
OH 1010 J=1:I=0:K=0:XLEN=0:OPTIO
        N BASE 1:' DECLARE ALL V
        ARIABLES USED
LP 1020 DIM ABCD%(1),ABC$(1),AB!
        (2,1),A#(1): 'DECLARE SA
        MPLE ARRAYS
DH 1030 AD!=PEEK(&H35A)+256*PEEK
        (&H35B)
JO 1040 PRINT TAB(2);"INTEGER";T
        AB(18);"STRING";TAB(32);
        "SINGLE PREC.";TAB(46);"
        DOUBLE PREC."
CH 1050 WHILE AD!<PEEK(&H35C)+25
        6*PEEK(&H35D)-1:XLEN=PEE
        K(AD!+3)
JP 1060 FOR I=0 TO XLEN+5+PEEK(A
        D!+XLEN+4)+256*PEEK(AD!+
        XLEN+5)
CF 1070 IF I>XLEN+5 THEN COLOR 0
        ,7 ELSE COLOR 7,0
OL 1080 IF I>XLEN+6+2*PEEK(AD!+X
        LEN+6) THEN COLOR 7,0
CA 1090 K=PEEK(AD!+I)
ON 1100 LOCATE I+5,15*(J-1)+1:PR
        INT USING "###    ";I;K;
FE 1110 NEXT I
LP 1120 AD!=AD!+XLEN+6+PEEK(XLEN
        +AD!+4)+256*PEEK(XLEN+AD
        !+5):J=J+1
FN 1130 WEND
OL 1140 LOCATE 19,1:PRINT "VARIA
        BLES:":PRINT TAB(11);"AB
        CD%(1)"
CO 1150 PRINT TAB(11);"ABC$(1)"
GB 1160 PRINT TAB(11);"AB!(2,1)"
NN 1170 PRINT TAB(11);"A#(1)"
KH 1180 PRINT "ARRAY DIMENSION A
        ND ELEMENTS PER DIMENSIO
        N ";CHR$(219);CHR$(219);
IE 1190 COLOR 7,0
EE 1200 LOCATE 1,79
```

Program 3: Variable Lister

```
DJ 10 REM PROGRAM ANALYZES DATA
      AREA AND PROVIDES A REPORT
       ON
DB 20 REM  1. SCALAR VARIABLES
NE 30 REM  2. ARRAYS
```

```
KP 40 REM  3. AMOUNT OF MEMORY R
      EQUIRED FOR VARIABLES
II 50 REM INTERNAL VARIABLES: AD
      0',8AS,CTV,CTA,II1,JJ1,XLE
      N
CL 60 REM                ARRAYS: IT
      V',VARNAME$,ARRNAME$,TMPNA
      ME$
MH 70 DEF SEG:SCREEN 0:WIDTH 80:
      COLOR 7,0:KEY OFF:CLS:ON E
      RROR GOTO 920
DB 80 LOCATE 1,1:PRINT "SORTING
      VARIABLES ";
GG 90 BAS=1:DIM TESTARRAY(2):TES
      TARRAY(0)=1:' TEST FOR OPT
      IDN BASE
CH 100 ERASE TESTARRAY
NK 110 AD0'=0:XLEN=0:CTV=-BAS:CT
      A=-8AS:II1=0:JJ1=0:DIM IT
      V'(10-8AS):' DECLARE INTE
      RNAL VAR.
ED 120 AD0'=PEEK(&H35B)+256*PEEK
      (&H359): 'COUNT NUMBER OF
       VARIABLES
MD 130 CTV=CTV+1:XLEN=PEEK(AD0'+
      3)
BJ 140 AD0'=AD0'+PEEK(AD0')+XLEN
      +4
DI 150 IF AD0'<PEEK(&H35A)+256*P
      EEK(&H35B)-1 GOTO 130
DN 160 AD0'=PEEK(&H35A)+256*PEEK
      (&H35B): ' COUNT NUMBER O
      F ARRAYS
FB 170 CTA=CTA+1:XLEN=PEEK(AD0'+
      3)
FH 180 AD0'=AD0'+XLEN+PEEK(AD0'+
      4+XLEN)+256*PEEK(AD0'+5+X
      LEN)+6
HM 190 IF AD0'<PEEK(&H35C)+256*P
      EEK(&H35D)-1 GOTO 170
GF 200 DIM VARNAME$(CTV),ARRNAME
      $(CTA): 'DECLARE STRINGS
      FOR VAR. AND ARRAYS
JH 210 GOSUB 590: 'GET ADDRESS O
      F INTERNAL VARIABLES
DD 220 CTV=-8AS:AD0'=PEEK(&H358)
      +256*PEEK(&H359)
BE 230 XLEN=PEEK(AD0'+3):FOR II1
      =1-BAS TO 7-8AS:IF AD0'=I
      TV'(II1) GOTO 280
KL 240 NEXT II1:CTV=CTV+1
IF 250 VARNAME$(CTV)=STR$(PEEK(A
      D0'))+CHR$(PEEK(AD0'+1)):
      'GET TYPE AND NAME
KM 260 IF PEEK(AD0'+2)>0 THEN VA
      RNAME$(CTV)=VARNAME$(CTV)
      +CHR$(PEEK(AD0'+2)):FOR I
      I1=1 TO XLEN:VARNAME$(CTV
      )=VARNAME$(CTV)+CHR$(PEEK
      (AD0'+3+II1)-12B):NEXT II
      1
IB 270 IF PEEK(AD0')=3 THEN VARN
      AME$(CTV)=VARNAME$(CTV)+"
      $"
BC 280 AD0'=AD0'+PEEK(AD0')+XLEN
      +4
GB 290 IF AD0'<PEEK(&H35A)+256*P
      EEK(&H35B)-1 GDTO 230
PA 300 CTA=-8AS:AD0'=PEEK(&H35A)
      +256*PEEK(&H35B)
IP 310 XLEN=PEEK(AD0'+3):FOR II1
      =B-8AS TO 10-BAS:IF AD0'=
      ITV'(II1) GOTO 410
YE 320 NEXT II1:CTA=CTA+1
DK 330 ARRNAME$(CTA)=STR$(PEEK(A
      D0'))+STR$(PEEK(AD0'+6+XL
      EN))+CHR$(PEEK(AD0'+1)):'
      GET TYPE AND NAME
OE 340 IF PEEK(AD0'+2)>0 THEN AR
      RNAME$(CTA)=ARRNAME$(CTA)
      +CHR$(PEEK(AD0'+2)):FOR I
      I1=1 TO XLEN:ARRNAME$(CTA
      )=ARRNAME$(CTA)+CHR$(PEEK
      (AD0'+3+II1)-12B):NEXT II
      1
DC 350 IF MID$(ARRNAME$(CTA),2,1
      )="3" THEN ARRNAME$(CTA)=
      ARRNAME$(CTA)+"$"
LL 360 ARRNAME$(CTA)=ARRNAME$(CT
      A)+"("
KJ 370 FOR II1=PEEK(AD0'+XLEN+6)
       TO 1 STEP -1
AA 380 ARRNAME$(CTA)=ARRNAME$(CT
      A)+STR$(PEEK(AD0'+6+XLEN+
      2*II1-1)+256*PEEK(AD0'+6+
      XLEN+2*II1)-8AS)
AE 390 IF II1=1 THEN ARRNAME$(CT
      A)=ARRNAME$(CTA)+")" ELSE
       ARRNAME$(CTA)=ARRNAME$(C
      TA)+","
EJ 400 NEXT II1
FM 410 AD0'=AD0'+XLEN+PEEK(AD0'+
      4+XLEN)+256*PEEK(AD0'+5+X
      LEN)+6
KL 420 IF AD0'<PEEK(&H35C)+256*P
      EEK(&H350)-1 GOTO 310
AG 430 ITV'(1-BAS)=PEEK(&H35A)+2
      56*PEEK(&H35B)-PEEK(&H35B
      )-256*PEEK(&H359)-64
GI 440 ITV'(2-BAS)=PEEK(&H35C)+2
      56*PEEK(&H350)-PEEK(&H35A
      )-256*PEEK(&H358)-102-3*(
      CTA+CTV+2*8AS)
EN 450 FOR II1=1-BAS TO CTA:ARRN
      AME$(II1)=RIGHT$(ARRNAME$
      (II1),LEN(ARRNAME$(II1))-
      1)
HM 460 JJ1=INSTR(1,ARRNAME$(II1)
      ,CHR$(32)):IF JJ1>0 THEN
      ARRNAME$(II1)=LEFT$(ARRNA
      ME$(II1),JJ1-1)+MID$(ARRN
      AME$(II1),JJ1+1,LEN(ARRNA
      ME$(II1))-JJ1)
EH 470 IF JJ1>0 GOTO 460
DP 480 NEXT II1:IF CTV=-1 THEN C
      TV=0 ELSE IF CTV=0 THEN C
      TV=1
FG 490 CLS:DIM TMPNAME$(CTV):FOR
       II1=1-BAS TO CTV:TMPNAME
      $(II1)=VARNAME$(II1):NEXT
       II1
KJ 500 JJ1=CTV-1:GDSUB B20:II1=0
      :GOSUB B60:GOSUB 940:ERAS
      E TMPNAME$:CLS:IF CTA=-1
      THEN CTA=0 ELSE IF CTA=0
      THEN CTA=1
DG 510 DIM TMPNAME$(CTA):FOR II1
      =1-8AS TO CTA:TMPNAME$(II
      1)=ARRNAME$(II1):NEXT II1
CD 520 JJ1=CTA-1:GOSUB B20:II1=1
      :GOSUB 860:GOSUB 940:ERAS
      E TMPNAME$,VARNAME$,ARRNA
      ME$:CLS
GD 530 PRINT "STRING SPACE USED
      ",TAB(30);
JM 540 AD0'=FRE(""):AD0'=PEEK(&H
      30A)-PEEK(&H32F)-1+256*(P
      EEK(&H30B)-PEEK(&H330)):P
      RINT USING "######";AD0'
FL 550 PRINT "SCALAR VARIABLE SP
       ACE",TAB(30);:PRINT USING
       "######";ITV'(1-BAS)
BH 560 PRINT "ARRAY SPACE",TAB(3
      0);:PRINT USING "######";I
      TV'(2-8AS)
DM 570 PRINT:PRINT "TOTAL VARIAB
      LE SPACE ",TAB(30);:PRINT
       USING "######";ITV'(1-BAS
      )+ITV'(2-8AS)+AD0';:PRINT
       " BYTES"
DP 580 GOSUB 940:CLS:END
DJ 590 ITV'(1-BAS)=VARPTR(AD0')-
      5:'SUBROUTINE TO GET INTE
      RNAL ADDRESSES
NF 600 ITV'(2-BAS)=VARPTR(XLEN)-
      6
JI 610 ITV'(3-8AS)=VARPTR(CTV)-5
MC 620 ITV'(4-8AS)=VARPTR(CTA)-5
GL 630 ITV'(5-8AS)=VARPTR(II1)-5
KI 640 ITV'(6-8AS)=VARPTR(JJ1)-5
HH 650 ITV'(7-8AS)=VARPTR(BAS)-5
DJ 660 ITV'(8-BAS)=VARPTR(ITV'(1
      -8AS))-10
MI 670 ITV'(9-8AS)=VARPTR(VARNAM
      E$(1-BAS))-14
BF 680 ITV'(10-8AS)=VARPTR(ARRNA
      ME$(1-8AS))-14
JD 690 FOR II1=1-8AS TO 9-BAS:IF
       ITV'(II1)<0 THEN ITV'(II
      1)=2^16+ITV'(II1)
DI 700 NEXT II1:RETURN
DD 710 XLEN=0:FOR II1=1-BAS TO C
      TV
HP 720 IF MID$(TMPNAME$(II1),1,2
      )=STR$(JJ1) AND MID$(TMPN
      AME$(II1),3,1)<"a" THEN P
      RINT MID$(TMPNAME$(II1),3
      ,LEN(TMPNAME$(II1))-2)+SP
      ACE$(2);:XLEN=1
DK 730 IF MID$(TMPNAME$(II1),1,2
      )=STR$(JJ1) AND MID$(TMPN
      AME$(II1),3,1)>"a" THEN P
      RINT "FN"+CHR$(ASC(MID$(T
      MPNAME$(II1),3,1))-128)+M
      ID$(TMPNAME$(II1),4,LEN(T
      MPNAME$(II1))-2)+SPACE$(2
      );:XLEN=1
EE 740 NEXT II1
BL 750 IF XLEN=0 THEN PRINT "NON
      E":PRINT:PRINT ELSE PRINT
      :PRINT
NN 760 RETURN
KP 770 FOR II1=1-BAS TO CTA
FC 780 IF LEFT$(ARRNAME$(II1),1)
      =RIGHT$(STR$(JJ1),1) THEN
       PRINT MID$(ARRNAME$(II1)
      ,3,LEN(ARRNAME$(II1))-2)+
      SPACE$(2);:XLEN=1
FO 790 NEXT II1
AC 800 IF XLEN=0 THEN PRINT "NON
      E":PRINT:PRINT ELSE PRINT
      :PRINT
EK 810 XLEN=0:RETURN
KE 820 XLEN=1:WHILE XLEN:XLEN=0
NM 830 FOR II1=1-BAS TO JJ1
LK 840 IF TMPNAME$(II1)>TMPNAME$
      (II1+1) THEN SWAP TMPNAME
      $(II1),TMPNAME$(II1+1):XL
      EN=1
OI 850 NEXT II1:WEND:RETURN
KM 860 IF II1=0 THEN PRINT "VARI
      ABLE SPACE ";ITV'(1-BAS);
      "8YTES":TMP$="" ELSE PRIN
      T "ARRAY SPACE ";ITV'(2-B
      AS);"8YTES  OPTION BASE "
      ;ABS(1-BAS);:TMP$="ARRAYS
      "
DL 870 PRINT:PRINT:PRINT:PRINT "
      INTEGER "+TMP$+" - ";:JJ1
      =2:IF TMP$="" THEN GOSUB
      710 ELSE GOSUB 770
BN 880 PRINT:PRINT:PRINT "STRING
       "+TMP$+" - ";:JJ1=3:IF T
      MP$="" THEN GOSUB 710 ELS
      E GOSUB 770
ME 890 PRINT:PRINT:PRINT "SINGLE
       PRECISION "+TMP$+" - ";:
      JJ1=4:IF TMP$="" THEN GOS
      UB 710 ELSE GOSUB 770
DM 900 PRINT:PRINT:PRINT "DOUBLE
       PRECISION "+TMP$+" - ";:
      JJ1=8:IF TMP$="" THEN GOS
      UB 710 ELSE GOSUB 770
MF 910 RETURN
CI 920 IF ERR=9 AND ERL=90 THEN
      BAS=0:RESUME NEXT
DL 930 ON ERROR GOTO 0
CN 940 LOCATE 25,1:PRINT "PRESS
      ANY KEY TO CONTINUE";
LJ 950 KB$=INKEY$:IF KB$="" GOTO
       950 ELSE RETURN
```

Apple IIc RAM Disk Mover Part 2

Christopher J. Flynn

Last month, Part 1 of this two-part series demonstrated the RAM disk and subdirectory options with ProDOS and the Apple IIc. This month's article presents a utility program which rapidly copies a number of programs from a floppy disk to the RAM disk, greatly speeding up the preparations required for using the RAM disk.

The discussion in Part 1 was a bit on the theoretical side, but this month we tackle the practical side. "RAM Disk Mover" is a program which automates the processes described in Part 1. It adds a few little twists, however. Here is how RAM Disk Mover works:

1. First it looks for the PROGRAMS subdirectory on your floppy disk. If PROGRAMS is not found, RAM Disk Mover instructs you to insert another disk.

2. Next, it looks in the RAM drive for the PROGRAMS subdirectory. If PROGRAMS is not found, RAM Disk Mover creates the subdirectory. If PROGRAMS is found, it deletes all of the files in PROGRAMS. This makes room for the new programs.

3. RAM Disk Mover goes back to the floppy PROGRAMS subdirectory. Then it saves the name and length (in blocks) of each BASIC program. It stops when there are no more BASIC programs or when the number of blocks exceeds the capacity of the RAM drive (118 blocks, considering the directories).

4. Using this list of BASIC programs, RAM Disk Mover builds an EXEC file containing a series of LOAD and SAVE commands. The EXEC file is named TEMP.EXEC and is stored in the volume directory of the floppy with which RAM Disk Mover is currently working.

5. RAM Disk Mover adds a RUN command as the last line of the EXEC file. The program specified by the variable P1$ will start automatically when the copy operation is finished. Right now, P1$ is set up for STARTUP.RAM. You can change this to whatever program name you want.

6. After building the EXEC file, RAM Disk Mover clears the screen and informs you that all is well. Then the EXEC file starts up. At this point, programs are actually moved from floppy disk to RAM disk. Your startup program will run and you'll be in business.

Preparing The Mover

RAM Disk Mover requires little, if any, of your attention while it runs. Your biggest job is to organize your disks so that Mover can access them properly.

Here are some guidelines for trouble-free operation:

Format some disks so you'll always have some spares handy. Use whatever volume labels suit you.

As you format the disks, be sure to establish the PROGRAM and DATA subdirectories.

Place a copy of Mover in the volume directory of each disk. Mover does not require much space.

Place all of the BASIC programs you want Mover to copy in the PROGRAMS subdirectory. Don't forget that the RAM disk has a limit of about 61K or 120 blocks. If you think you will exceed this limit, place some of the programs on a second disk.

Putting It To Work

Now for the actual operation. It's simple:

1. Always make sure your IIc is turned on, that ProDOS is ready, and that you're in Applesoft.

2. Place your program disk in the internal disk drive.

3. Type either RUN MOVER or -MOVER. RAM Disk Mover will take over from there. It tells you the name of each program it is copying, then starts the EXEC file. As the EXEC file runs, you will see a series of open brackets displayed. This means all is well.

4. When the EXEC file is done, the STARTUP.RAM program will begin running if it is present.

5. At this point, you can remove your floppy disk from the disk drive. You can insert a data disk and have the entire 140K free for data storage.

Now What?

RAM Disk Mover has done its job. The BASIC programs have been moved over to the RAM disk. Now, how do you get to the programs?

If you want to run a program, you can type:

RUN /RAM/PROGRAMS/*program-name*
or
- /RAM/PROGRAMS/*program-name*

Perhaps typing all this seems a bit tedious. You can use the PREFIX command:

PREFIX /RAM/PROGRAMS

Now whenever you want to run a program, you can just type RUN followed by the program name. /RAM/PROGRAMS/ is automatically appended to the front of the name.

A caution is in order, however. Suppose that your program performs some file operations. If the input/output statements do not include a full path name, the prefix will also be applied to your data file commands. This will cause your program to try to read or write to the RAM disk—probably not what you intended. Remember this when using the PREFIX command.

You can also use the usual LOAD and SAVE commands, of course. But this time a warning is in order. Yes, you can recall a program from the RAM disk, work on it, test the revisions, and store it back in the RAM disk. *Just don't forget it's only a RAM disk.* If the power goes kaplooey, then your program goes kaplooey along with it. If you make important changes to a program,

save the new version on a real live disk. The RAM disk is best for programs you just want to run.

Oops!

On occasion things can go wrong—especially where computers are concerned. Here are a few gotchas. Watch out for them.

• RAM Disk Mover is designed for an Apple IIc and ProDOS. It uses the 80-column display capability of the IIc. If the display looks funny, make sure your computer is in the 80-column mode.

• Mover attempts to use all of the available RAM drive space. It will clean up /RAM/PROGRAMS, but it is not aware of anything else that you may have in the RAM drive. If you run out of room, ProDOS will tell you about it loudly and clearly.

• There can be problems writing the EXEC file. Do not remove your program disk until RAM Disk Mover is finished with it. Do not write-protect it, either. Finally, leave a few blocks free for Mover's use. That EXEC file has to go somewhere.

Additional Hints

Programs can run other programs. If the programs are in the RAM disk, switching from program to program is almost instantaneous. Here's an example:

```
10 REM PROGRAM1
20 D$=CHR$(4)
30 PRINT "PROGRAM 1 IS RUNNING"
40 PRINT D$;"-RAM/PROGRAMS/PROG
   RAM2"
```

```
10 REM PROGRAM2
20 D$=CHR$(4)
30 PRINT "PROGRAM 2 IS RUNNING"
40 PRINT D$;"-RAM/PROGRAMS/PROG
   RAM1"
```

How about that! Keep this technique in mind when you are writing that huge program that eats all the available program space. The way out is to think small and think RAM drive. ProDOS even has a CHAIN statement that permits variables to be passed between programs.

If you would like to learn more about ProDOS, find a copy of Apple's *BASIC Programming with ProDOS*. It covers all the ProDOS features available from Applesoft BASIC.

Apple IIc RAM Disk Mover

```
100 REM MOVE /.../PROGRAMS TO /
    RAM/PROGRAMS (MOVER)
110 REM
120 PF$ = "/RAM/PROGRAMS/"
130 P1$ = "STARTUP.RAM"
140 RB = 11B: REM /RAM BLOCKS A
    VAILABLE
150 MP = 50: REM MAX PROGRAMS
160 DIM PN$(MP)
170 HOME :D$ = CHR$ (4): PRINT
    D$;"PR#3"
180 GOSUB 330: REM TITLE
190 GOTO 400: REM OPEN DIRECTOR
    Y
200 GOTO 650: REM CREATE /RAM D
    IRECTORY
210 GOSUB B60: REM MOVE PROGRAM
    S
220 REM
230 REM THE EXEC PROGRAM WILL M
    OVE THE PROGRAMS
240 REM
250 HOME
260 VTAB 12: HTAB 1: PRINT "Run
    ning the EXEC copy program
    ... "
270 FOR I = 1 TO 1000: NEXT
280 PRINT D$;"EXEC ";FL$
290 HOME
300 VTAB 12: HTAB 1: PRINT "Pro
    grams are being copied to:"
    ;PF$
310 FOR I = 1 TO 1000: NEXT
320 END
330 REM PROGRAM TITLE
340 UL$ = "": FOR I = 1 TO 80:U
    L$ = UL$ + "_": NEXT
350 TL$ = "DISK TO /RAM PROGRAM
     MOVER"
360 HOME : VTAB 1: HTAB 1: PRIN
    T UL$
370 VTAB 3: HTAB (80 - LEN (TL$
    )) / 2: PRINT TL$
380 VTAB 4: HTAB 1: PRINT UL$
390 RETURN
400 REM OPEN /.../PROGRAMS DIRE
    CTORY
410 GOSUB 590: REM GET VOLUME L
    ABEL
420 DR$ = VL$ + "PROGRAMS/"
430 VTAB 6: HTAB 1: PRINT "Movi
    ng programs from ";DR$
440 ONERR GOTO 480
450 PRINT D$;"OPEN ";DR$;",TDIR
    "
460 POKE 216,0: REM NORMAL ERR
470 GOTO 200
480 POKE 216,0: REM NORMAL ERR
490 CALL - 32BB: REM FIX STACK
500 VTAB 8: HTAB 1: PRINT "The
    directory '/PROGRAMS/' is n
    ot on the diskette ";VL$
510 VTAB 10: HTAB 1: PRINT "Ins
    ert the proper diskette."
520 VTAB 12: HTAB 1: PRINT "Pre
    ss SPACE to continue.  Pres
    s RETURN to stop."
530 GET C$: IF C$ < > " " AND C
    $ <  >  CHR$ (13) THEN 530
540 FOR I = 6 TO 12 STEP 2
550 VTAB I: HTAB 1: PRINT SPC(
    79); CHR$ (13)
560 NEXT
570 IF C$ = CHR$ (13) THEN STOP
580 GOTO 400
590 REM GET VOLUME LABEL
600 PRINT D$;"PREFIX /"
610 PRINT D$;"PREFIX"
620 INPUT VL$
630 PRINT D$
640 RETURN
650 REM CREATE /RAM/PROGRAMS/
660 VTAB 8: HTAB 1: PRINT "Movi
    ng programs to ";PF$
670 ONERR GOTO B20
680 PRINT D$;"OPEN ";PF$;",TDIR
    "
690 PRINT D$;"READ ";PF$
700 INPUT L1$
710 INPUT L2$
720 INPUT L3$
730 INPUT L4$: IF L4$ = "" THEN
     770
740 T$ = MID$ (L4$,2,15)
750 NP = NP + 1:PN$(NP) = T$
760 GOTO 730
770 PRINT D$;"CLOSE ";PF$
780 FOR I = 1 TO NP
790 PRINT D$;"DELETE ";PF$;PN$(
    I)
800 NEXT
810 GOTO 210
820 POKE 216,0: REM DIRECTORY N
    OT PRESENT
830 CALL - 32BB: REM FIX STACK
840 PRINT D$;"CREATE ";PF$
850 GOTO 210
860 REM MOVE PROGRAMS
870 NP = 0: REM NUMBER OF PROGR
    AMS
880 BC = 0: REM BLOCK COUNT
890 PRINT D$;"READ ";DR$
900 INPUT L1$: REM DIRECTORY NA
    ME
910 INPUT L2$: REM TITLE LINE
920 INPUT L3$: REM BLANK LINE
930 INPUT L4$: REM FILE ENTRY
940 IF L4$ = "" THEN 970
950 GOSUB 1000: REM SAVE PGM NA
    MES
960 GOTO 930
970 PRINT D$;"CLOSE"
980 GOSUB 1160: REM BUILD EXEC
    FILE
990 RETURN
1000 REM SAVE PGM NAMES
1010 IF MID$ (L4$,1B,3) < > "BA
     S" THEN 1150
1020 NP = NP + 1
1030 IF NP > MP THEN 1150
1040 BC = BC + VAL ( MID$ (L4$,
     23,6))
1050 IF BC > R8 THEN 1150
1060 T$ = MID$ (L4$,2,15)
1070 FOR K = 15 TO 1 STEP - 1
10B0 IF MID$ (T$,K,1) < > " " T
     HEN 1100
1090 NEXT
1100 PN$(NP) = LEFT$ (T$,K)
1110 IF PN$(NP) = P1$ THEN SU =
      1
1120 VTAB 10: HTAB 1: PRINT SPC
     ( 79); CHR$ (13)
1130 VTAB 10: HTAB 1: PRINT "Co
     pying ";PN$(NP);" ..."
1140 FOR K = 1 TO 500: NEXT
1150 RETURN
1160 REM BUILD EXEC FILE
1170 VTAB 10: HTAB 1: PRINT SPC
     ( 79); CHR$ (13)
1180 VTAB 10: HTAB 1: PRINT "Bu
     ilding EXEC copy program .
     .."
1190 FL$ = "TEMP.EXEC"
1200 PRINT D$;"OPEN ";FL$
1210 PRINT D$;"CLOSE ";FL$
1220 PRINT D$;"DELETE ";FL$
1230 PRINT D$;"OPEN ";FL$
1240 PRINT D$;"WRITE ";FL$
1250 FOR I = 1 TO NP
1260 PRINT "LOAD ";VL$;"PROGRAM
     S/";PN$(I)
1270 PRINT "SAVE ";PF$;PN$(I)
1280 NEXT
1290 IF SU = 1 THEN PRINT "-";P
     F$;P1$
1300 PRINT "PRINT:PRINT"
1310 PRINT D$;"CLOSE"
1320 RETURN
```

Commodore Disk Editor

Martin Sikes

Examine or edit any sector on a disk with this short, useful program for the Commodore 64 and 1541 disk drive. Recommended for intermediate to advanced users.

One of the best ways to learn about disk storage is to look at the actual contents of sectors on the disk. "Commodore Disk Editor" lets you do just that and even change the contents if you wish. Equipped with this tool, you can repair garbled disks, retrieve accidentally scratched files, as well as protect your programs against tampering by others.

To make the most of Disk Editor, you'll need to know at least the basics of Commodore disk storage. Consult your *1541 User's Manual.* Additional information also is available in various books and programming guides. Even without advanced knowledge, however, you can perform some interesting tricks. A few examples are given below.

To get started, type in the program and save it. When running it for the first time, *put an unimportant disk in the drive.* Making mistakes with a program like this can scramble a disk and render it useless. Until you have gained some experience, practice on an unimportant disk. Make a backup copy of any important disk before you try to alter it.

Reading And Editing Blocks

Disk Editor begins by prompting you to enter track and sector numbers of the block of data you want to examine. Commodore 1541 disks have 35 tracks, and each track contains a number of sectors. (The terms *sector* and *block* are often used interchangeably to describe a 256-byte data area on the disk.) The available track and sector numbers are listed below. Tracks 1–17 each have 21 sectors, numbered 0–20, and so on.

Tracks	Sectors
1–17	0–20
18–24	0–18
25–30	0–17
31–35	0–16

To choose a particular sector, you must enter two numbers separated by a comma. The program does not let you enter nonexistent track or sector values, with the exception noted below. For example, enter 18,1 and press RETURN to examine track 18, sector 1 (this block holds the first part of the disk directory). Disk Editor reads that block from the disk and displays its contents at the top of the screen.

Disk Editor displays the block using the upper-/lowercase character mode. Thus, the values on the screen correspond to the numbers in your *64 User's Guide* screen code table: A zero is displayed as an @ character, 13 as a lowercase m, and so on.

The next prompt asks whether you want to edit the block displayed on the screen. Press Y to edit, or N to choose another sector. If you select Y, press the cursor keys to move the blinking cursor atop the character you want to change. Then type in the new character over the old one. For instance, if the first filename in your directory is DOG, you can change the name to BOG simply by typing SHIFT-B over the D.

When your editing is complete, you again have two options. You can press RETURN to write the changed block back to the disk, or press the f1 special function key to escape the editing mode without changing anything on the disk. In either case, you'll return to the first prompt, so you can read the same sector again to verify your changes or proceed to another sector. To exit the program, enter track 99, sector 99 at the first prompt. These are the only nonexistent track and sector numbers that Disk Editor allows you to enter.

Don't Boggle The BAM

You've already seen how to change filenames by editing directory blocks. The disk name can be changed just as easily, by editing track 18, sector 0. Note that these names cannot be more than 16 characters long.

Sector 0 of track 18 contains the *Block Availability Map,* or BAM—important information about the organization of programs on the disk. It's very easy to damage files by improperly changing things in this sector. Until you're familiar with the BAM, avoid changing anything in this sector other than the disk name.

You may notice that a copy of the disk ID follows the disk name in the BAM. However, you *cannot* change the disk ID by changing this value in the BAM. (You can change the value of the ID shown in sector 0, but this won't change the disk ID; there's no way to do that without reformatting the disk.)

Hidden Programs

Lots of tricks are possible with Disk Editor. Perhaps you'd like to conceal a program's name to make it difficult for others to find in the

directory. Directory lists are displayed on the screen with the equivalent of PRINT, so it's easy to disrupt the list by inserting control characters in a filename.

For instance, you might conceal the filename DOG by adding three delete characters after the name. The ASCII table in your *64 User's Guide* tells you that a delete character has a value of 20. This corresponds to lowercase *t* in the screen code table, so you simply type three *t*'s after DOG. Now the program appears as "" when you list the directory. But you can still load the program with the filename DOG* or DOG???. It can also be loaded with the line A$="DOG" + CHR$(20) + CHR$(20) + CHR$(20): LOAD A$,8.

Most characters are easily entered from the keyboard. To enter a reverse video character, press CTRL-9 before typing the character. Disable reverse video by pressing CTRL-0.

Like many protection methods, of course, these simple tricks are only effective against people who know less than you do. It's not very difficult to write a BASIC program that displays a directory without using PRINT statements. If your friends also are familiar with Disk Editor, your attempts at concealment will be utterly transparent. Nevertheless, such methods should be sufficient to deter casual users.

As your knowledge grows, you'll find more practical uses for Disk Editor, such as changing the file type to prevent a program from being scratched. You can also copy blocks from one disk to another, store data directly in unused disk sectors, and restore damaged files to their original condition.

Commodore Disk Editor

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
10 REM BLOCK EDIT          :rem 214
20 FORX=0TO26:READA:POKE820+X,
   A:NEXT                  :rem 28
30 DATA 162,2,32,198,255,160,0
   ,32,207                 :rem 16
40 DATA 255,153,0,4,169,1,153,
   0,216                   :rem 170
50 DATA 200,208,242,162,1,32,1
   98,255                  :rem 228
60 DATA 96                 :rem 239
70 PRINT"{CLR}{CYN}":POKE53281
   ,0:POKE53280,9          :rem 2
80 POKE 53272,23:PRINTCHR$(8)
                           :rem 124
90 OPEN 1,0:OPEN15,8,15
                           :rem 240
100 GOSUB380               :rem 172
110 INPUT"ENTER TRACK, SECTOR"
   ;T,S                    :rem 227
120 IFT=99THEN470          :rem 239
130 IFT<1ORS>20ORT>35THEN100
                           :rem 211
140 IFT>17ANDT<25ANDS>18THEN10
   0                       :rem 117
150 IFT>24ANDT<31ANDS>17THEN10
   0                       :rem 112
160 IFT>30ANDS>16THEN100
                           :rem 166
170 OPEN 2,8,2,"#"         :rem 81
180 PRINT#15,"U1:"2;0;T;S
                           :rem 153
190 SYS820:CLOSE2:GOSUB420:GOS
   UB380                   :rem 186
200 INPUT"EDIT AND SAVE? {RVS}
   Y{OFF}ES OR {RVS}N{OFF}O";
   A$:IFA$<>"Y"THEN100
                           :rem 105
220 POKE254,PEEK(1024):GOSUB38
   0:PRINT"{WHT}";         :rem 103
230 PRINT"USE THE CURSOR TO TY
   PE OVER THE BLOCK.
   {2 SPACES}IF YOU MAKE A MI
   STAKE, DO ";            :rem 160
240 PRINT"NOT PRESS{5 SPACES}R
   ETURN. PRESS THE F1 FUNCTI
   ON KEY."                :rem 85
250 POKE781,0:POKE782,0:SYS655
   20                      :rem 99
260 POKE204,0:WAIT198,255:WAIT
   207,1:POKE1024,PEEK(254)
                           :rem 100
280 GETA$:IFPEEK(203)=0THEN280
                           :rem 233
290 IFA$=CHR$(13)THENWAIT207,1
   :POKE204,1:GOTO340      :rem 66
300 IFA$=CHR$(133)THEN330
                           :rem 114
310 POKE205,3:WAIT207,1:PRINTA
   $;:GOTO260              :rem 144
330 CLOSE1:CLOSE15:OPEN15,8,15
   ,"I0":CLOSE15:PRINT"{CLR}
   {2 DOWN}OK":GOSUB460:GOTO7
   0                       :rem 162
340 OPEN2,8,2,"#":PRINT"{HOME}
   {14 DOWN}{RVS}SAVING TRACK
   "T"SECTOR"S"            :rem 136
350 FORX=0TO254:PRINT#2,CHR$(P
   EEK(1025+X));:NEXT:PRINT#2
   ,CHR$(PEEK(1024));:rem 163
360 PRINT#15,"U2:"2;0;T;S:CLOS
   E2                      :rem 124
370 A$="":GOSUB420:GOTO100
                           :rem 212
380 PRINT"{HOME}{6 DOWN}"
                           :rem 229
390 PRINTCHR$(5);:FORX=0TO39:P
   RINTCHR$(162);:NEXT:PRINTC
   HR$(159)                :rem 254
400 FORX=1TO100:PRINT"
   {5 SPACES}";:NEXT       :rem 52
410 PRINT"{HOME}{7 DOWN}":RETU
   RN                      :rem 10
420 INPUT#15,A,B$,C,D      :rem 141
430 IF B$="OK"THEN RETURN
                           :rem 182
440 PRINTA;B$;C;D:POKE53281,9
                           :rem 213
450 FORX=1TO2000:NEXT:POKE5328
   1,0:GOTO100             :rem 240
460 FORX=1TO2000:NEXT:RETURN
                           :rem 63
470 PRINT"OK":CLOSE1:PRINT#15,
   "I0":CLOSE15:END :rem 71 ©
```

Lyco Computer Marketing & Consultants

SAVE ON THESE IN STOCK PRINTERS

AXION
GP 550 AT (Atari)	249
GP 550 CD (C 64)	249
GP 550 PC (IBM)	239
GP 550 AP (Apple)	279
GP 700 AT (Atari)	459
GP 700 AP (Apple)	459
Elite 5CD (C-64)	329

BLUE CHIPS
M12010	$275
M12010 C 64	$275

C. ITOH
Prowriter 8510 AP	279
8510 BC2	389
8510 BP1	319
8510 SP	379
8510 SR	429
8510 SCP	459
8510 SCR	479
7500 AP	205
7500 AP	245
1550 P	449
1550 BCD	489
A-10-20-P	459
F 10 40 PU or RDU	888
F10 SSPU or RDU	1069

CARDCO
LQ1	369
LQ3	279

CITIZEN
MSP-10	329
MSP 15	499
MSP-20	479
MSP-25	649

COMREX
CR-II EC Comrter IIE Parallel	359
CR-II ES Comrter II E Parallel	379
CR IV C Comrter IV Parallel	689
CR IV S Comrter IV Serial	689

Corona
LP300 Laser Printer	2699
200361 Toner Cartridge	89

DIGITAL DEVICES
16K printer buffer	99 75
32K printer buffer	119 75
64K printer buffer	169 95

EPSON
RX-80	225
RX 80 FT+	279
FX100+	579
JX 80	529
LQ 1500 P	1089
LQ 1500 S	1149
HI 80 Color Plotter	399

JUKI
Juki 6100	379
RS 232 Serial Board	55
Tractor	119
Sheet Feeder	209
Juki 6300	769

LEGEND
880	219
1080	239
1200	249

★ PRINTER INTERFACING Available ★

MANNESMANN TALLY
Spirit 80	255
MTL-160L	549
MTL-180L	739

NEC
NEC 8025	$699
NEC 8027	$359

OKIDATA
Okimate 10	179
82A	295
84	645
92	349
93	565
92 Imagewriter	425
92 IBM Version	349

OLIVETTI
DY 250 Parallel	739
DY 250 Serial	729
DY 450 Parallel	1099
DY 450 Serial	1079

PANASONIC
1090	189
1091	259
1092	395
1093	589
3151	459

Smith Corona
Fastext 80	189 00
D100	219 00
D200	399 00
D300	519 00
L1000	339 00

STARMICRONICS
SG-10	219
SG-15	379
SD-10	339
SD-15	445
SR-10	489
SR 15	585
Powertype	309
Gemini 10X	CALL
Gemini 15X	CALL
SB 10	CALL

OVER 2000 SOFTWARE TITLES IN STOCK

"BRAND NEW/FACTORY FRESH MERCHANDISE!" "WE OFFER FULL LINE OF ACCESSORIES!"

COMPUTER CARE

BIB
DISK DRIVE CLEANER	$12 75
COMPUTER CARE KIT	$19 75

NORTRONICS DISK DRIVE CLEANER with software for IBM-PC, Atari, Vic,
DISK DRIVE CLEANER with software for IBM-PC Atari Vic Apple TI	$29 75
DISK CLEANER REFILL	$14 75
CASSDRIVE CLEANER	$9 95
MEDIA BULK ERASER	$46 75

NEC
PC8201 Portable	$429
NECB1 64K Computer System	$1049
NECB2 128 K Computer System	$1299
PC8221 Thermal Printer	$ 139
PC8201 8K RAM Chip	$ 99
PC8206 32K RAM Cart	$ 299
PC300 Modem	$ 65
PC8801 MSDOS 16 Bit Card	$ 339

PRINTING PAPER

3000 SHEETS FANFOLD	$42 75
1000 SHEETS FANFOLD	$19 75
1000 SHEET LETTER	$21 95
200 SHEETS LETTER	$8 99
150 RAG STATIONARY	$10 99
MAILING LABELS (1in)	$9 95
14 x 11 1000 FANFOLD	$24 75

INNOVATIVE CONCEPTS
Flip n File 10	3 50
Flip n File 15	8 25
Flip-n File 25 Lock	17 95
Flip n File 50	17 25
Flip n File 50 Lock	22 95
Flip-n File Rom	17 25

DRIVES

MSD
SD1 Drive	229
SD2 Drive	469

INDUS
GT Atari 219
GT Atari	219
GT Commodore	249
GT Apple w/controller	219
GT Apple	169

DISKETTES

SKC (Box 10)
5¼ SKC SSSD	10 99
5¼ SKC SSDD	13 99
5¼ SKC-DSDD	15 99

ELEPHANT (Box 10)
5¼ SSSD	13 99
5¼ SSDD	15 99
5¼ DSDD	19 99

MAXELL
5¼ MD-1	16 99
5¼ MD 2	23 75

(Box 10)

IBM-PC SOFT-WARE

*LOTUS
Lotus 1-2 3	309 00
Symphony	439 00

*ASHTON-TATE
Framework	385 00
d Base II	289 00
d Base III	385 00
d Base II upgrade	135 00

*PARADISE
5-Pak Multifunction	179 00
Modular Graphics	289 00
Modular Graphics Card	315 00

IBM-PC COMPATABLE

CORONA
PPC 22A Portable 256K Amber	1F99
PPC 22G Portable 256-K Green	1699
PPCXTA Portable 256K 10Meg CORI 28K 128K RAM	2899 1599

Zenith
Z-150	Call

Leading Edge
PC Compatable	Call

*LEADING EDGE
Nutshell	69 95
LEWP Basic	65 00
LEWP Merge Print	99 00
LE Spell Correction	169 00

*QUADRAM
Quad Jr Exp Chassis	519 00
Quad Jr Exp Memory	209 00
Quad Memory Jr	209 00
Quadcolor I	209 00

*STB
Super R10-64K	279 00
Super R10 192K	399 00
Super R10 256K	449 00
Graphics Plus II	339 00

*PARADISE
Multi-Display Card	335 00

CONTINENTAL
Home Accountant	85 00

TOLL FREE 1-800-233-8760

MasterCard TO ORDER VISA

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

or send order to
Lyco Computer
P O Box 5088
Jersey Shore PA 17740

RISK FREE POLICY

In stock item shipped within 24 hours of order No deposit on C O D orders Free shipping on prepaid cash orders within the Continental U S PA residents add sales tax APO FPO and International orders add $5 00 plus 3% for priority mail service Advertised prices show 4% discount for cash add 4% for Master Card or Visa Personal checks require 4 weeks clearance before shipping All items subject to change without notice

For your protection, we check for stolen credit cards.

ATARI | apple | COMMODORE

ATARI

800 XL Computer	CALL
1050 Drive	169
1010 Recorder	44
1020	55
1025	185
1027	239
850	109

SYNAPSE
(ATARI)

Synfile	34 95
Syncalc	34 95
Syncomm	27 95
Syntrend	27 95
Synchron	27 95
Synstock	27 95

SCARBOROUGH

Net Worth	49 95
Mastertype	24 75
Improved Mastertype	24 75
Mastertype s Filer	24 75

SPINNAKER

Delta Drawing Room	19 95
Cosmic Life Room	19 95
Up for Grabs Room	19 95

SSI

Baseball	22 75
Question	26 75
50 Mission Crush	22 75
Broadsides	22 75
Computer Ambush	34 75

Trillium

Shadowkeep	$26 75
Fahrenheit 451	$26 75
Amazon	$26 75

MICROPOSE

Solo Flight	22 75
NATO	22 75
Spitfire Ace	19 95
F-15 Strike Eagle	22 75

CONTINENTAL

Home Accountant	44 75
1985 Book of Atatn Software	16 95

SUBLOGIC

Flight Simulator II	32 75
Night Mission Pinball	18 75

PERSONAL PERIPHEIALS

Super Sketch Atari	32 95

BRODERBUND

Bank St Writer	42 75
The Print Shop	29 95
Serpent's Star	24 75
Spelunker	19 95
Stealth	19 95

BUSINESS

VISICALC	$159 75
LETTER PERFECT R ..	59 00
DATA PERFECT	$89 75
FILE MANAGER	$69 75
HOME FILE MGR	$69 75

ADVENTURE

Dishey	29 95
Ultra Disassembler	29 95

GRAPHIC TABLET

Super Sketch	32 95
Koala Pad	59 95

EASTERN HOUSE

Monkey Wrench II800	24 95
Monkey Wrench II800 XL	24 95

CONTINENTAL

Home Accountant	44 95
1985 Book of Alan Software	16 95

ADVENTURE INTERNATIONAL

Diskey	32 75
Ultra Dissambler	32 75

WICO
(Joysticks)

15-9714 Bat Handle	16 99
50-2002 Super 3-way	19 99
72 4545 Trackball	29 99

Scarborough

Songwriter	$24 75
Picturewnt	$24 75
Phi Beta F	$32 75
Mastertype	$24 75
Run f Money	$32 75
Net Worth	$54 75

Microprose

Solo Flight	$22 75
NATO	$22 75
Spitfire	$22 95
F 15 Strike	$22 75
Air Rescue	$22 75

SSI

Baseball	$22 75
Questron	$26 75
Germany 1985	$32 75
50 Missions	$21 75

Spinnaker

Alphabet	$18 75
Story Machine	$19 75
Kids on Keys	$18 75
Grandma	$19 75
Snooper Troop	$22 75

Broderbund

Bank St Writer	$42 75
Bank St Filer	$42 75
Bank St Mailer	$42 75
Bank St Spell	$42 75
Mask of Sun	$24 95
Print shop	32 95
Lode Runner	$22 95

Graphics Tablet

Supersketch	$49 95
Koala	$84 95

COMMODORE

Simon s Basic	24 75
Assembler 64	34 75
Super Expander	22 75
Logo 64	49 75
Pilot 64	38 75
Easy Cale	34 75
Easy Script	38 75
C 64 Computer	CALL
C 1541 Disk Drive	195
MPS 801 Printer	175
C 1702 Monitor	209
C 1531 Datasette	39 75
C 1660 Auto Modem	85

SS1
(C 64)

Computer Baseball	24 75
Field of Fire	24 75
Computer Quarterback	24 75
Questron	24 75
50 Mission Crush	24 75

Scarborough

Songwriter	$24 75
Phi Beta F	$29 95
Mastertype	$24 75
Run f Money	$24 75
Net Worth	$49 95

SUBLOGIC
(C 64)

Flight Simulator II	32 75
Night Mission Pinball	22 75

PERSONAL PERIPHERALS

Super Sketch 64	32 75
Printer Utility	18 75

KOALA
(C 64)

Koala Pad	59 95

CARDCO

C/01 Write Now	29 95
C/02 Write Now - 64	39 95
D/01 Mail Now - 64	29 00
D/04 Spell Now - 64	29 00
D/02 Utility Desk	19 95
CSD 1 Disk Drive (new)	CALL
MOD 1 Modern (new)	CALL
D/03 Tax Payer (new)	27 95
D/07 CalcNow/64 (new)	27 95
D/08 SUper Printer Utility	
D/08 Super Printer Utility	27 95
CK/1 Numeric Key Pad	34 95
DC/1 Data Cassette	39 95
CB/5 5 Slot Board C-64	54 00
CR/1 Light Pen	29 75
CE/1 Cassette Interface	29 75
CB/3 3 Slot Board Vic-20	24 95
CB/6 6 Slot Board Vic-20	65 00

HES

HES Games 64	22 95
Omni Writer/Spell	34 95
HES Mon 64	23 95
Microsoft Multiplan	55 00
Type N Write	19 95
Turtle Graphics II	23 95
Cell Delense	22 95
Paint Brush	12 95
Tri Math	22 95
Graphics Basic	27 95
HES Kit	29 95
Millionaire	23 95
64 Forth	24 95
HES Writer 64	24 95

Timaworks

Inventory	$32 75
Sales	$32 75
Accts Rec	$32 75
Accts Rec	$32 75
G Ledger	$39 75
Data Mgr	$14 75
Checkbook	$14 75
Star Battle	$14 75
Cave of Word	$18 75

Microprose

Solo Flight	$22 75
NATO	$22 75
Spitfire	$19 95
F 15 Strike	$22 75
Air Rescue	$22 75

MODEMS

MICROBITS

MPP 1000 E (Atari)	99 00
MPP 1064 (C 64	69 95

HAYES

Smartmodem 300	189
Smartmodem 1200	459
Smartmodem 1200B	389
Micromodem IIE	249
Micromodem 100	289
Chronograph	179
Smart Com II	75

TELE LEARNING

CM 250 (C 64)	55 00
AP-250 (Apple)	109 95
IB 250 (IBM)	109 95

CARDCO MOD 1 (C 64)	CALL
NESTRIDGE (C 64)	CALL
MITEY MO (C 64)	CALL
1660 AUTO MODEM (C 64)	85
COMPUSERVE	23 95

★ *LOWEST PRICES!* ★ ★ *SUPER SPECIAL!* ★

MONITORS

AMDEK

300 Green	125
300 Amber	139
310 Amber IBM	155
Color 300 Audio	245
Color 500 Composite	369
Color 600	429
Color 700	495
Color 710	569

PANASONIC

DT 1300 RG1 composite	329

SAKATA

SC 100 Color	219
STS1 Stand	29
SG 1000 Green	99
SA 1000 Amber	109

PRINCETON GRAPHICS

MAX 12 Amber	189
HX-12 RGB	475
SR 12 RGB	599

NEC

JB-1260 Green	95
JB-1201 Green	135
JC 1215 Color	235
JC 1216 RGB	375
JC1460 Color	265
JB-1205 Amber	139

GORILLA

12 Green	78
2 Amber	84

ZENITH

ZVM 122A Amber	84
ZVM 123G Green	75
ZVM 124 Amber IBM	129
ZVM 131 Color	275
ZVM 133 RGB	389
ZVM 135 Composite	449
ZVM 136 Hi Res Color	589

TAXAN

210 Color RBG	249
115 Green	119
116 Amber	125
400 Color RGB	275
410 Color RGB	339
420 Color IBM	429
121 Green IBM	139
122 Amber IBM	145

X-TRON

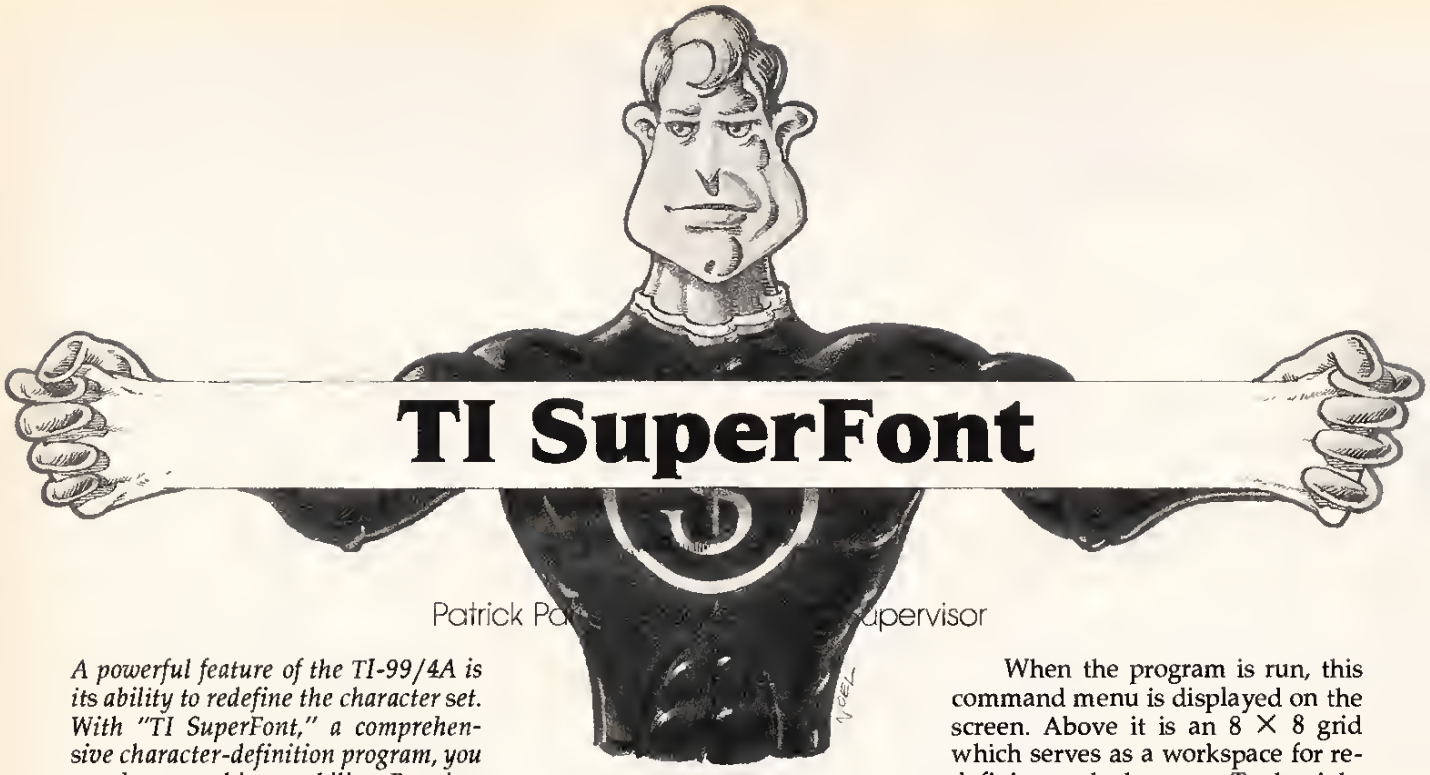
Comcolor I Composite Green	199

AMERICA'S MAIL ORDER HEADQUARTERS

LYCO COMPUTER

WORLD'S LEADER IN SALES & SERVICE

TO ORDER
CALL TOLL FREE
800-233-8760
In PA 1 717-327-1824
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740



TI SuperFont

Patrick Pa[illegible] upervisor

A powerful feature of the TI-99/4A is its ability to redefine the character set. With "TI SuperFont," a comprehensive character-definition program, you can harness this capability. Requires Extended BASIC and a joystick (printer optional).

The character graphics capabilities of the TI-99/4A are well known. But to redefine a character on the TI by the usual means (see the *TI User's Reference Guide*, pages II-76 to II-79), you must follow a tedious, multistep procedure. First, you plot the character in an 8 × 8 grid. Next, you convert each row of the grid into a two-digit hexadecimal number and then sequentially combine the numbers from each row to generate a *pattern identifier*, or coded representation of the character. Finally, you place this pattern identifier along with a chosen ASCII value for the character in a CALL CHAR statement. Anyone who has repeatedly endured this process can attest to its drudgery.

Fortunately, the process is easily computerized, and several character-definition programs have been written for the TI. Until now, however, these programs have not taken full advantage of the TI's capabilities. With "TI SuperFont" (Program 1), once-tedious character manipulations can now be undertaken with ease.

Sixteen Commands

SuperFont, originally written for the Atari by Charles Brannon, first appeared in the January 1982 issue of COMPUTE! and featured 18 commands for redefining characters. After using this outstanding program on several occasions, I was convinced that TI users deserved a utility as versatile and convenient. That's how TI SuperFont was born.

In converting SuperFont, a few commands with less value to the TI user were eliminated while certain more practical commands were added. As it turned out, the real challenge was to fit the program into a TI without memory expansion. The final version leaves only a hundred or so bytes to spare. However, certain adjustments must be made if you are using a disk drive with the program. Before loading and running SuperFont, type CALL FILES(1). This will free up memory ordinarily reserved for additional disk file manipulation.

TI SuperFont offers the following 16 commands or modes:

E	EDIT	N	INPUT
R	RESTORE CH	H	RESTORE CHSET
F	COPY	W	WRITE DATA
M	MIRROR	V	REVERSE
A	ROTATE	C	CLEAR
I	INSERT	D	DELETE
L	LOAD FONT	S	SAVE FONT
P	PRINT CH	T	PRINT CHSET

When the program is run, this command menu is displayed on the screen. Above it is an 8 × 8 grid which serves as a workspace for redefining each character. To the right of the grid, the current mode and, in some cases, a prompt will be displayed. Below this is printed the entire TI character set (codes 32–143) with each subset (eight characters) denoted by a different background color. (If you find the colors annoying, remove the FOR-NEXT loop in line 300.)

Several commands require that you pick a character from the TI character set. In these instances, a box-shaped sprite (CHR$(143)) appears over the last character referenced from the set (defaults to space). Position the sprite with the joystick over the desired character and press the fire button. Unless indicated otherwise, each command returns you to the *EDIT* mode upon completion.

Now let's examine each command, beginning with *EDIT*. (The ALPHA-LOCK key should be up when making menu selections.)

- *EDIT* is the basic editing command. After you press E, SuperFont requests you to choose a character from the character set. The character selected is copied into the grid and the box-shaped sprite appears. This is actually like a cursor, controlled with the joystick. Press the fire button to set a point (if a point is clear) or reset a point (if a point is already set). You can draw lines by holding down the button while moving the joystick. When you're pleased with

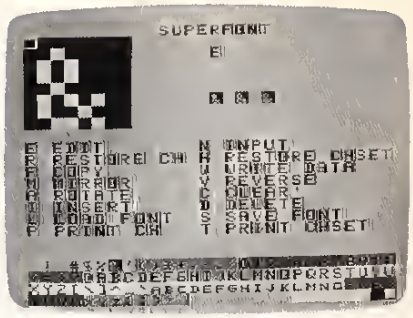
the appearance of the shape in the grid, press ENTER to redefine the character. (To completely redesign a character from scratch, use the *CLEAR* command, described below.)

- *INPUT* lets you type in a pattern identifier and assign it to a particular character code. After selecting *INPUT*, choose a character from the set with the joystick and then type in the hexadecimal code for the redefined character. The hexadecimal code can be typed in upper- or lowercase (a routine at line 960 automatically converts the code to uppercase). The *INPUT* command is handy when attempting to associate a pattern identifier with a CHR$ code in someone else's program.
- *RESTORE CH* restores the current character to its original configuration. This command is useful if you've mangled a character or changed the wrong one.
- *RESTORE CHSET* restores the entire character set to its initial appearance.
- *COPY* copies a character to a second location in the character set. SuperFont prompts you for the first character (the one to be moved) and the second character (the destination character). This command is handy for arranging your customized characters to fit the various color codes.
- *WRITE DATA* displays the pattern identifier for each selected character along with its ASCII value. Very handy when comparing characters or for providing a few character codes for another program.
- *MIRROR* produces a mirror image of the current character in the grid.
- *REVERSE* puts the current character in the grid in reverse field: All dots become blanks, and all blanks become dots.
- *ROTATE* turns the current character 90 degrees clockwise.
- *CLEAR* completely clears out the current character. For creating new characters from scratch.
- *INSERT* places a row of blanks in the current character. Move the cursor in the grid with the joystick to the row where you wish to insert the blanks and press ENTER. All rows below will scroll down and the bottom row will be lost.
- *DELETE* is the opposite of *INSERT*. Position the cursor on a row in the grid and press ENTER. The row will be eliminated and all other rows will scroll upward. *DELETE* and *INSERT* can be used with *ROTATE* to scroll characters left or right in the grid (of course, one row will be lost in both cases).
- *LOAD FONT* loads a previously SAVEd character set (a font) from tape or disk. SuperFont prompts you for the device and filename. Be sure to type this in the standard format (that is, CS1 or DSK1.FILENAME). Again, capital letters need not be used. The routine that converts from lower- to uppercase takes care of this for you. If you're using tape, the screen will be restored after the tape system messages have been printed (the same occurs with *SAVE FONT*, discussed below). When loading is complete, a command prompt appears.
- *SAVE FONT* saves to tape or disk (in a data file format) only those characters which have been altered since SuperFont was run. Since each character code is saved as a separate record, a large set may take 30 minutes to save. As with *LOAD FONT*, you will be prompted for the device and filename. If you accidentally hit L (for *LOAD FONT*) or S from the main menu, simply press ENTER to abort the command when prompted for the device and filename.

Once saved, character sets can be loaded into any program where they're needed (we'll consider this in greater detail shortly). As with *LOAD FONT*, a command prompt appears when the operation is complete.

- *PRINT CH* prints the current character in an 8 × 8 grid along with its ASCII and pattern identifier codes, then returns you to the main menu. Be sure to modify line 1260 to correspond to the specifications of your printer.
- *PRINT CHSET* is the same as the previous command, except it prints every character which has been modified.

The commands offered by SuperFont are versatile, but you may want to add others. Since the program uses most of the TI's memory, unless you have additional RAM you'll have to substitute your own routine for an existing one. Fortunately, the program is modular in structure. Just follow the branching IF statements from line 360 to 920 for the current commands. If you do alter the program, test it thoroughly to make sure you still have plenty of memory left.



Redesigning a character with "TI SuperFont."

Retrieving A Font Or Screen

After you've saved a newly created character set, how do you go about recovering it for use in another program? Program 2 shows how this is done.

In line 130, the device and filename for the character set file is defined as B$ (the filename used here is FONT). If you store this file on tape rather than disk, line 130 should read B$="CS1". Lines 140–160 load in the new character set and print it on the screen. Line 170 sets up a delay so you can see that the character set has successfully loaded.

With SuperFont, you can perform many chores with ease. You can customize your character set (ever wished for true lowercase?), create graphics characters and animated figures (space creatures!), or just play around. The uses of this utility are endless. I'm sure you'll have as much fun discovering them as I have.

Program 1: TI SuperFont

```
10 DIM A$(111),C$(15),N$(11
   2),D(15),V(8,8):: L=32
20 E=15 :: Q$="DEVICE.FILEN
   AME?" :: GOSUB 1240 :: G
   OTO 260
30 F=0 :: GOSUB 40 :: GOTO
   340
40 CALL HCHAR(5,14,L,16)::
   RETURN
50 CALL HCHAR(3,17,L,7):: C
   ALL HCHAR(7,17,L,16):: R
   ETURN
60 FOR I=5 TO 7 :: CALL HCH
```

```
   AR(I,13,L,1B):: NEXT I :
   : RETURN
70 Z$=N$(W-L)
80 FOR I=0 TO 15 :: O(I)=AS
   C(SEG$(Z$,I+1,1))-48 ::
   O(I)=O(I)+(O(I)>9)*7
90 NEXT I :: J=0 :: FOR I=0
    TO 7 :: DISPLAY AT(2+I,
   1):C$(O(J));:: DISPLAY A
   T(2+I,5):C$(O(J+1));:: J
   =J+2 :: NEXT I :: RETURN
100 CALL DELSPRITE(#1):: DI
    SPLAY AT(5,15):"WAIT"
110 FOR R=1 TO B :: FOR C=1
    TO B
120 IF M=109 THEN CALL GCHA
    R(R+1,11-C,H):: GOTO 15
    0
130 IF M=97 THEN CALL GCHAR
    (10-C,R+2,H):: GOTO 150
140 CALL GCHAR(R+1,2+C,H)
150 V(R,C)=H-141 :: NEXT C
    :: NEXT R
160 H$="01234567B9ABCDEF" :
    : IF M=11B THEN H$="FED
    CBA9B76543210"
170 Z$="" :: FOR R=1 TO B :
    : LD=V(R,5)*B+V(R,6)*4+
    V(R,7)*2+V(R,B)+1
1B0 HI=V(R,1)*B+V(R,2)*4+V(
    R,3)*2+V(R,4)+1
190 Z$=Z$&SEG$(H$,HI,1)&SEG
    $(H$,LO,1):: NEXT R
200 IF (M<>100)*(M<>105)THE
    N 240
210 IF M<>100 THEN 230
220 Z$=SEG$(Z$,1,ROW*2-2)&S
    EG$(Z$,ROW*2+1,14)&"00"
     :: GOTO 240
230 Z$=SEG$(Z$,1,RDW*2-2)&"
    00"&SEG$(Z$,RDW*2-1,16-
    ROW*2)
240 CALL CHAR(W,Z$):: N$(W-
    L)=Z$ :: IF (M=100)+(M=
    105)THEN GOSUB 70
250 GOSUB 40 :: RETLRN
260 F$="0000000100100011010
    0010101100111100010011O
    1010111100110111101111"
270 FOR I=0 TO 15 :: Z$=SEG
    $(F$,I*4+1,4):: O$=""
280 FOR J=1 TO 4 :: T=VAL(S
    EG$(Z$,J,1))+141 :: D$=
    O$&CHR$(T):: NEXT J ::
    C$(I)=D$ :: NEXT I
290 CALL CHAR(141,"",142,RP
    T$("F",16),143,"FFB1B1B
    1B1B1B1FF"):: FOR I=141
     TO 143 :: CALL CHARPAT
    (I,A$(I-L)):: N$(I-L)=A
    $(I-L):: NEXT I
300 CALL DELSPRITE(#1):: CA
    LL CLEAR :: FOR I=2 TO
    14 :: CALL COLOR(I,2,I+
    2):: NEXT I
310 FOR I=L TO 143 :: PRINT
     CHR$(I);:: NEXT I :: D
    ISPLAY AT(1,11):"SUPERF
    ONT" :: GOSUB 1100
320 FOR R=1 TO B :: CALL HC
    HAR(R+1,3,141,B):: NEXT
     R
330 BR=20 :: BC=2 :: W=L
340 CALL SOUND(100,B00,2)::
     DISPLAY AT(3,15):"MODE
    ?"
350 CALL KEY(0,M,S):: IF S=
    0 THEN 350
360 IF M<>101 THEN 510
370 T=1 :: GOSUB 1200 :: GO
    SUB 9B0 :: IF (F=1)*(K<
    >112)THEN 30 ELSE IF K=
    112 THEN M=K :: GOSUB 4
    0 :: GOTO 910
3B0 GOSUB 70 :: Z=1
390 CALL SPRITE(#1,143,10,9
    ,17):: R=1 :: C=2 :: CA
    LL GCHAR(R+1,C+1,T)
400 CALL KEY(0,K,S):: IF (K
    =13)+(K=112)THEN ROW=R
    :: GOSUB 100 :: GOSUB 1
    1B0 :: IF K<>112 THEN O
    N Z GOTO 340,590
410 IF (K>13)THEN M=K :: GO
    TO 360
420 CALL JOYST(1,X,Y):: IF
    ABS(X)+ABS(Y)=B THEN 42
    0
430 CALL KEY(1,KK,S):: IF (
    KK<>1B)*(ABS(X)+ABS(Y)=
    0)THEN 400
440 OK=0 :: IF ABS(X)+ABS(Y
    )=4 THEN OK=1
450 C=C-(X=4)+(X=-4):: R=R-
    (Y=-4)+(Y=4)
460 C=C-(C=1)*B+(C=10)*B ::
     R=R-(R=0)*B+(R=9)*B
470 CALL LOCATE(#1,B*R+1,B*
    C+1)
480 IF (KK=1B)*(DK=0)THEN C
    ALL GCHAR(R+1,C+1,T)::
    T=2B3-T
490 IF (OK=1)*(KK<>1B)THEN
    CALL GCHAR(R+1,C+1,T)
500 CALL HCHAR(R+1,C+1,T)::
     CALL SOUND(-1,294,3)::
     GOTO 400
510 IF M<>110 THEN 570
520 T=1 :: GOSUB 1200 :: GO
    SUB 9B0 :: IF F=1 THEN
    30
530 DISPLAY AT(5,12):"CHAR
    HEX CODE?" :: ACCEPT AT
    (6,11)SIZE(16)BEEP:O$ :
    : IF LEN(O$)<>16 THEN 5
    30
540 GOSUB 60 :: GOSUB 960
550 N$(W-L)=Z$ :: GOSUB B0
    :: CALL CHAR(W,Z$)
560 GOSUB 40 :: GOTO 590
570 IF M<>114 THEN 600
580 GOSUB 1200 :: CALL CHAR
    (W,A$(W-L)):: N$(W-L)=A
    $(W-L)
590 Z=1 :: GOSUB 40 :: GOSU
    B 70 :: M=101 :: GOSUB
    11B0 :: CALL HCHAR(3,17
    ,69):: CALL SOUND(50,BB
    0,3):: GOTO 390
600 IF M<>104 THEN 620
610 GOSUB 1200 :: FOR I=L T
    O 143 :: CALL CHAR(I,A$
    (I-L)):: N$(I-L)=A$(I-L
    ):: NEXT I :: GOTO 590
620 IF M<>102 THEN 670
630 GOSUB 1200
640 DISPLAY AT(5,15):"1ST C
    HAR?" :: GOSUB 9B0 :: I
    F F=1 THEN 30 ELSE TM=W
650 GOSUB 70 :: DISPLAY AT(
    5,15):"2NO CHAR?" :: GO
    SUB 9B0 :: IF F=1 THEN
    30 ELSE CALL DELSPRITE(
    #1)
660 CALL CHARPAT(TM,Z$):: C
    ALL CHAR(W,Z$):: N$(W-L
    )=Z$ :: GOTO 590
670 IF M=109 THEN GOSUB 120
    0 :: GOSUB 100 :: GOTO
    590
6B0 IF M=11B THEN GOSUB 120
    0 :: GOSUB 100 :: GOTO
    590
690 IF M<>97 THEN 730
700 GOSUB 1200
710 GOSUB 100 :: GOSUB 70 :
    : GOSUB 11B0 :: T=0 ::
    D$="AGAIN (Y/N)?" :: GO
    SUB 1220 :: GOSUB 40 ::
     IF T=1 THEN 710
720 GOTO 590
730 IF M=99 THEN GOSUB 1200
     :: O$=RPT$("0",16):: C
    ALL CHAR(W,O$):: N$(W-L
    )=D$ :: GOTO 590
740 IF M=105 THEN GOSUB 120
    0 :: Z=2 :: GOTO 390
750 IF M=100 THEN GOSUB 120
    0 :: Z=2 :: GOTO 390
760 IF M<>119 THEN B20
770 T=1 :: GOSUB 1200 :: GO
    SUB 9B0 :: IF F=1 THEN
    F=0 :: GOTO B10 ELSE GO
    SUB 70
7B0 DISPLAY AT(7,16):"CHAR=
    ";W :: DISPLAY AT(9,11)
    :N$(W-L)
790 O$="AGAIN(Y/N) ?" :: GO
    SUB 1220
B00 CALL HCHAR(9,11,L,1B)::
     IF T=1 THEN GOSUB 40 :
    : GOTO 770
B10 GOSUB 60 :: GOTO 340
B20 IF M<>10B THEN B60
B30 GOSUB 1200
B40 GOSUB 940 :: OPEN #1:D$
    ,INTERNAL,INPUT ,FIXED
B50 INPUT #1:T,N$(T):: IF T
    <>112 THEN CALL CHAR(T+
    L,N$(T)):: GOTO B50 ELS
    E CLOSE #1 :: GOSUB 60
    :: IF ASC(O$)=67 THEN 3
    00 ELSE 340
B60 IF M<>115 THEN 910
B70 GOSUB 1200 :: GOSUB 940
B80 OPEN #1:O$,INTERNAL,OUT
    PUT,FIXED :: FOR I=L TO
     143
B90 IF N$(I-L)<>A$(I-L)THEN
     PRINT #1:I-L,N$(I-L)
900 NEXT I :: T=112 :: F$="
    " :: PRINT #1:T,F$ :: C
    LOSE #1 :: GOSUB 60 ::
    IF ASC(O$)=67 THEN 300
    ELSE 340
910 IF M=112 THEN H=1 :: GO
    SUB 1260
920 IF M=116 THEN H=0 :: GO
    SUB 1260
930 GOTO 340
940 DISPLAY AT(5,13):Q$ ::
    ACCEPT AT(6,14):O$ :: I
    F O$="" THEN GOSUB 60 :
    : GOTO 340 ELSE GOSUB 9
    60
950 RETURN
960 Z$="" :: FOR I=1 TO LEN
    (O$):: F$=SEG$(D$,I,1):
    : IF (ASC(F$)>96)*(ASC(
    F$)<123)THEN F$=CHR$(AS
    C(F$)-L)
970 Z$=Z$&F$ :: NEXT I :: O
    $=Z$ :: RETURN
9B0 CALL SPRITE(#1,143,10,B
    R*B+1,BC*B+1)
990 CALL JOYST(1,X,Y):: IF
    ABS(X)+ABS(Y)=B THEN 99
    0
1000 BC=BC-(X=4)+(X=-4):: W
     =W-(X=4)+(X=-4)
1010 BR=BR-(Y=-4)+(Y=4):: W
     =W-(Y=-4)*2B+(Y=4)*2B
1020 IF BC<2 THEN BC=29 ::
     BR=BR-1
1030 IF BC>29 THEN BC=2 ::
     BR=BR+1
1040 IF BR<20 THEN BR=23 ::
      W=W+112
1050 IF BR>23 THEN BR=20 ::
      W=W-112
1060 CALL KEY(1,KK,ST):: CA
     LL KEY(0,K,S)
```

```
1070 IF S<>0 THEN F=1 :: IF
     M=111 THEN RETURN ELS
     E CALL DELSPRITE(#1)::
     RETURN
1080 IF KK=1B THEN CALL SOU
     NO(10,110,2):: GOSUB 4
     0 :: CALL DELSPRITE(#1
     ):: RETURN
1090 GDTD 980
1100 DISPLAY AT(11,1):"E EO
     IT";TAB(14);"N INPUT"
1110 DISPLAY AT(12,1):"R RE
     STDRE CH";TA8(14);"H R
     ESTDRE CHSET"
1120 OISPLAY AT(13,1):"F CO
     PY";TAB(14);"W WRITE D
     ATA"
1130 DISPLAY AT(14,1):"M MI
     RRDR";TAB(14);"V REVER
     SE"
1140 OISPLAY AT(15,1):"A RO
     TATE";TA8(14);"C CLEAR
     "
1150 DISPLAY AT(16,1):"I IN
     SERT";TAB(14);"D DELET
     E"
1160 DISPLAY AT(17,1):"L LD
     AD FONT";TAB(14);"S SA
     VE FDNT"
1170 DISPLAY AT(1B,1):"P PR
     INT CH";TA8(14);"T PRI
     NT CHSET" :: RETURN
1180 FDR I=0 TO 5 STEP 2 ::
      CALL HCHAR(7,17+I,W):
     : NEXT I :: RETURN
1190 R=20 :: C=2 :: W=L ::
     CALL SPRITE(#1,143,2,R
     *8+1,C*B+1):: RETURN
1200 GOSUB 50 :: CALL HCHAR
     (3,17,M-L):: IF T=1 TH
     EN OISPLAY AT(5,15):"P
     ICK A CHAR" :: T=0
1210 RETURN
1220 DISPLAY AT(5,15):D$ ::
      ACCEPT AT(5,27)BEEP V
     ALIDATE("yn")SIZE(1):Z
     $ :: IF Z$="y" THEN T=
     1
1230 RETURN
1240 CALL CLEAR :: CALL SCR
     EEN(E):: DISPLAY AT(12
     ,7):"LOADING CHARPATS"
      :: FOR I=127 TO 140 :
     : CALL CHAR(I,""):: NE
     XT I
1250 FOR I=L TO 140 :: CALL
      CHARPAT(I,A$(I-L)):
     N$(I-L)=A$(I-L):: NEXT
      I :: RETURN
1260 DISPLAY AT(3,15):"PRIN
     T" :: DPEN #1:"RS232/2
     .8A=9600.DA=B.PA=N"
1270 TM=W :: IF H=1 THEN 13
     00
12B0 FDR T=L TD 143 :: IF N
     $(T-L)<>A$(T-L)THEN W=
     T ELSE 1350
1290 E=E+1 :: E=(E=17)*14+E
     :: CALL SCREEN(E)
1300 IF ((F=1)*(H=1))+(H=0)
     THEN GDSUB 70 :: GDSU8
      1180
1310 FDR R=2 TD 9 :: IF R=5
      THEN PRINT #1:TAB(5);
     "CHR$ # - "&"<"&STR$(W
     )&">";
1320 PRINT #1:TAB(30);:: FO
     R C=3 TO 10 :: CALL GC
     HAR(R,C,X):: IF X=141
     THEN X=45 ELSE X=88
1330 PRINT #1:CHR$(X);:: NE
     XT C :: IF R=5 THEN PR
     INT #1:TA2(47);"HEX CO
     DE - "&"<"&N$(W-L)&">"
1340 NEXT R :: PRINT #1 ::
     PRINT #1 :: IF H=1 THE
     N 1360
1350 NEXT T
1360 CLDSE #1 :: F=0 :: H=0
      :: E=15 :: W=TM :: CA
     LL SCREEN(E):: RETURN
```

Program 2: Character Set Loader

```
100 'GAME
110 'GET REDEFINED CHARS
120 CALL CLEAR
130 B$="DSK1.FONT"
140 OPEN #1:B$,INTERNAL,INP
    UT ,FIXED
150 INPUT #1:F,NEWA$ :: IF
    F<>112 THEN CALL CHAR(F
    +32,NEWA$):: PRINT CHR$
    (F+32);:: GDTO 150
160 CLDSE #1
170 FOR T=1 TO 1000 :: NEXT
     T
```

Apple ProDOS Variable Lister

Paul F. Stuever

This fast machine language utility takes the pain out of debugging BASIC programs by listing the current value of every program variable. You can also make a hardcopy of the variable list. ProDOS is required.

How many times have you run a program, only to get a message like OVERFLOW ERROR IN 240, or worse yet, BAD SUBSCRIPT ERROR IN 240? When you list the line in question, it may look something like this:

```
240 A$(X +XZ,2 * (B / 4 = C + 1),B / 4)
= STR$ (Z)
```

To locate the error, you'll need to type PRINT X, followed by PRINT XZ and so on, to find the current value of each variable. This is a slow, tedious way to debug a program, especially when you find that some of these variables were defined with other formulas.

"Apple Variable Lister" takes the drudgery out of such debugging tasks by quickly listing the current value of every variable in your program. The program is written in machine language and works on any Apple II series computer with ProDOS.

You can use this utility even if you don't understand machine language: The BASIC loader program listed below creates the machine language and saves it on your disk. Type in the loader, and save a copy before you run it in case you made a typing error. The program has a checksum to catch errors and identify any lines that have mistakes. If no errors are found, it prints OK and saves the utility with the filename VAR.LIST on your disk as a binary file.

Once this is done, you're ready to use the lister. Enter BLOAD VAR. LIST to load it into memory, followed by HIMEM: 31000 to set the top of memory. You will ordinarily want to do this at the beginning of a programming session. To list your variables, simply type CALL 32000 and press RETURN. The same command can run the routine from within a BASIC program. To make a hardcopy of the variable list, enter PR#1 before calling the routine.

A Chronological List

Variable Lister displays your program's arrays first, followed by floating point, string, and integer variables. The variables are displayed in chronological order (the order they are used in the program), not alphabetically. Although Applesoft BASIC allows arrays with up to 88 dimensions and as many elements per dimension as available memory will allow, Variable Lister is more restrictive. For this program, arrays are limited to three dimensions and a maximum of 254 elements per dimension. Attempting to list a larger array—for example, the one created by DIM A$(500)—crashes the utility.

Note that Variable Lister cannot display a variable until it has actually been used in the program. For instance, consider the following line:

```
10 A$="YES":IF A$="NO" THEN
B$="OK"
```

Since the IF condition can never be satisfied, B$ will not appear on the variable list unless the program uses it elsewhere. This is no problem when debugging, since you're interested only in variables that were used up to the time the program crashed. However, to make a complete variable list for permanent documentation, you'll need to run your program until you know that every variable has been used.

Apple Variable Lister

```
100 HOME : PRINT "CREATING VAR.
    LIST": HIMEM: 31000
110 X = 32000:TC = 0: PRINT
120 Z = 0: FOR A = 0 TO 9
130 READ P: POKE X,P:Z = Z + P
140 X = X + 1: IF X > 32601 THE
    N 200
150 NEXT A:TC = TC + Z
160 READ A: IF A = Z THEN 120
170 PRINT "ERRDR IN DATA "
180 PRINT "CHECK LINE # ";X - 3
    1010: STOP
190 :
200 IF TC = 85238 THEN PRINT "O
    K": PRINT  CHR$ (4);"BSAVE
    VAR.LIST,A$7D00,L601": END
210 PRINT "ERRDR IN DATA "
220 PRINT "MISSING A LINE": STO
    P
1000 DATA 032, 127, 125, 169, 0
     00, 133, 004, 032, 017, 12
     5, 764
1010 DATA 230, 004, 032, 017, 1
     25, 230, 004, 169, 141, 03
     2, 984
1020 DATA 237, 253, 165, 106, 1
     33, 236, 165, 105, 208, 00
     9, 1617
1030 DATA 169, 007, 024, 101, 2
     35, 144, 002, 230, 236, 13
     3, 1281
1040 DATA 235, 165, 236, 197, 1
     08, 144, 009, 240, 001, 09
     6, 1431
1050 DATA 165, 235, 197, 107, 1
     76, 249, 032, 251, 126, 22
     8, 1766
1060 DATA 004, 208, 223, 032, 0
     19, 127, 160, 002, 166, 00
     4, 945
1070 DATA 240, 029, 202, 240, 0
     13, 169, 165, 032, 237, 25
     3, 1580
1080 DATA 032, 072, 249, 032, 2
     28, 126, 208, 198, 169, 16
     4, 1478
1090 DATA 032, 237, 253, 032, 1
     73, 126, 032, 188, 126, 20
     8, 1407
1100 DATA 185, 032, 072, 249, 1
     64, 236, 165, 235, 024, 10
     5, 1467
1110 DATA 002, 144, 001, 200, 0
     32, 249, 234, 032, 046, 23
     7, 1177
1120 DATA 169, 141, 032, 237, 2
     53, 208, 159, 169, 000, 13
     3, 1501
1130 DATA 004, 032, 141, 125, 2
     30, 004, 032, 141, 125, 23
     0, 1064
1140 DATA 004, 169, 141, 032, 2
     37, 253, 165, 107, 133, 23
     5, 1476
1150 DATA 165, 108, 208, 011, 1
     65, 237, 024, 101, 235, 13
     3, 1387
1160 DATA 235, 165, 238, 101, 2
     36, 133, 236, 197, 110, 24
     0, 1891
1170 DATA 003, 144, 007, 096, 1
     65, 235, 197, 109, 176, 24
     9, 1381
1180 DATA 160, 003, 177, 235, 1
     33, 238, 136, 177, 235, 13
     3, 1627
1190 DATA 237, 032, 251, 126, 2
     28, 004, 208, 212, 132, 25
     2, 1682
1200 DATA 132, 251, 132, 250, 1
     60, 004, 177, 235, 170, 20
     0, 1711
1210 DATA 200, 177, 235, 149, 2
     49, 202, 208, 247, 134, 25
     5, 2056
1220 DATA 134, 254, 134, 253, 1
     52, 056, 101, 235, 133, 23
     5, 1687
```

```
1230 DATA 169, 000, 101, 236, 1
     33, 236, 032, 019, 127, 16
     6, 1219
1240 DATA 004, 208, 005, 032, 0
     44, 126, 208, 011, 202, 20
     8, 1048
1250 DATA  005, 032, 074, 126,
     208, 003, 032, 102, 126, 1
     64, 872
1260 DATA 253, 166, 254, 165, 2
     55, 200, 196, 250, 144, 01
     6, 1899
1270 DATA 160, 000, 232, 228, 2
     51, 144, 009, 162, 000, 02
     4, 1210
1280 DATA 105, 001, 197, 252, 1
     76, 009, 132, 253, 134, 25
     4, 1513
1290 DATA 133, 255, 076, 236, 1
     25, 165, 236, 076, 167, 12
     5, 1594
1300 DATA 032, 127, 126, 165, 2
     35, 164, 236, 032, 249, 23
     4, 1600
1310 DATA 032, 046, 237, 169, 1
     41, 032, 237, 253, 024, 16
     9, 1340
1320 DATA 005, 101, 235, 144, 0
     02, 230, 236, 133, 235, 09
     6, 1417
1330 DATA 169, 164, 032, 237, 2
     53, 032, 127, 126, 160, 00
     0, 1300
1340 DATA 032, 173, 126, 152, 0
     56, 101, 235, 133, 235, 16
     9, 1412
1350 DATA 000, 101, 236, 133, 2
     36, 076, 191, 126, 169, 16
     5, 1433
1360 DATA 032, 237, 253, 032, 1
     27, 126, 160, 000, 032, 22
     8, 1227
1370 DATA 126, 024, 165, 235, 1
     05, 002, 144, 002, 230, 23
     6, 1269
1380 DATA 133, 235, 096, 169, 1
     68, 032, 237, 253, 165, 25
     3, 1741
1390 DATA 032, 034, 127, 165, 2
     51, 240, 024, 169, 172, 03
     2, 1246
1400 DATA 237, 253, 165, 254, 0
     32, 034, 127, 165, 252, 24
     0, 1759
1410 DATA 010, 169, 172, 032, 2
     37, 253, 165, 255, 032, 03
     4, 1359
1420 DATA 127, 169, 169, 032, 2
     37, 253, 076, 072, 249, 17
     7, 1561
1430 DATA 235, 133, 142, 200, 1
     77, 235, 133, 002, 200, 17
     7, 1634
1440 DATA 235, 133, 003, 096, 0
     32, 072, 249, 169, 162, 03
     2, 1183
1450 DATA 237, 253, 166, 142. 2
     40, 018, 160, 000, 177, 00
     2, 1395
1460 DATA 009, 128, 032, 237, 2
     53, 165, 241, 032, 168, 25
     2, 1517
1470 DATA 200, 202, 208, 240, 1
     69, 162, 032, 237, 253, 16
     9, 1872
1480 DATA 141, 076, 237, 253, 1
     77, 235, 133, 158, 200, 17
     7, 1787
1490 DATA 235, 133, 159, 162, 1
     44, 024, 032, 155, 235, 03
     2, 1311
1500 DATA 046, 237, 169, 141, 0
     76, 237, 253, 162, 000, 16
     0, 1481
1510 DATA 001, 177, 235, 016, 0
     01, 232, 009, 128, 133, 00
     1, 933
1520 DATA 136, 177, 235, 016, 0
     01, 232, 009, 128, 133, 00
     0, 1067
1530 DATA 096, 165, 000, 032, 2
     37, 253, 165, 001, 032, 23
     7, 1218
1540 DATA 253, 165, 241, 076, 1
     68, 252, 160, 000, 162, 00
     0, 1477
1550 DATA 201,  100, 144, 012, 1
     60, 176, 162, 176, 200, 05
     6, 1387
1560 DATA 233, 100, 201, 100, 1
     76, 248, 201, 010, 144, 01
     0, 1423
1570 DATA 162, 176, 232, 056, 2
     33, 010, 201, 010, 176, 24
     8, 1504
1580 DATA 009, 176, 072, 138, 0
     72, 152, 240, 003, 032, 23
     7, 1131
1590 DATA 253, 104, 240, 003, 0
     32, 237, 253, 104, 076, 23
     7, 1539
1600 DATA 253, 000, 000, 000, 0
     00, 000, 000, 000, 000, 00
     0, 253
1610 DATA  000, 000, 000, 000,
     000, 000, 000, 000, 000, 0
     00,  0
```

Atari Cassette Filenames

Norman Lin

Do you have trouble loading Atari cassette files because you keep losing track of the tape counter numbers? Or maybe you're wasting lots of tape by recording only one program per cassette side. Now there's a solution—a clever way to add filename capability to Atari cassettes. The technique works on any Atari 400/800, XL, or XE.

Unlike some other tape storage systems, the Atari doesn't allow filenames for cassette files. You must either jot down the tape counter numbers where the files start, or record only one file on each side of a cassette. But what happens if your cassette recorder's counter goes awry, or if you lose the index numbers? Things would be a lot easier if the computer could locate a program in the middle of a tape and load it for you.

Finally there's a simple way to solve these problems: "Atari Searcher/Loader." It lets you save numerous programs on a single side of a cassette, and then automatically finds and loads the program you want.

Saving Programs

Atari Searcher/Loader is very easy to use. Just follow these steps:

1. Type in the program listing following this article. (Note: Line 90 is too long to be typed as listed; to enter it, you must abbreviate POSITION as POS. When you list the program, POS. automatically appears as POSITION. Don't attempt to edit the line after it is entered. If you make a mistake, retype the entire line.)

2. Save the program once at the beginning of each tape using the LIST"C:" command—*not* the CSAVE command. (Just type LIST"C:", press RETURN, push the Play and Record buttons on the recorder, and hit RETURN again. Of course, you'll have to start with blank tapes to avoid overwriting programs on your existing tapes.) After you've saved Atari Searcher/Loader with LIST"C:", *do not rewind* the tape. Type NEW to clear Searcher/ Loader out of memory.

3. Enter the following short line in immediate mode (that is, without a line number):

```
OPEN#1,8,0,"C:":? #1;
"filename":CLOSE #1
```

where *filename* is the name you wish to assign to your program. Then press the Play and Record buttons and hit RETURN twice. After a few seconds, the filename is written onto tape and the computer's READY prompt reappears.

4. Now you can start saving your regular program as usual, except that you must use the LIST"C:" command as described above instead of CSAVE. If you want to load a program from another tape to save onto the Searcher/Loader tape with a filename, swap cassettes without rewinding the Searcher/Loader tape.

Repeat steps 3 and 4 for each program you save on that side of the tape.

The filename can be anything you like. Disk filenames are limited to eight characters plus a three-character extender (such as PROGRAM1.BAS), but Atari Searcher/ Loader permits much longer filenames. However, you should not include spaces or graphics characters as part of a name. Stick to letters, numbers, and common symbols. Do not use the same filename more than once on the same side of a cassette. It is a good idea to write the filenames on the cassette label in case you forget them.

Automatic Loading

Loading your programs with Atari Searcher/Loader is even easier than saving them. Suppose you've saved five programs on one tape using the above procedure. Their filenames are PROG1, PROG2, PROG3, PROG4, and PROG5. Now you want to load PROG4. Just follow these steps:

1. Rewind the tape to the beginning and load Atari Searcher/ Loader by typing this command and pressing RETURN:

```
ENTER"C:"
```

2. When the READY prompt reappears, type RUN. Searcher/ Loader asks, FILENAME?. Type in the filename (in this example, PROG4) and press RETURN.

Searcher/Loader hunts through the tape until it finds PROG4, then automatically loads it and stops.

How It Works

After Searcher/Loader asks you for the filename, it stores the name in the string variable A$ and enters the FOR-NEXT loop at lines 40–80. This loop searches for and loads one block of data at a time (made possible by the LIST"C:" format in which the programs are saved). Each block of data is stored in B$. If you'd like to see these blocks of data printed on the screen during the search process, insert line 65 PRINT B$.

Line 70 checks to see if B$ equals A$—in other words, if the block of data loaded is the same as the filename you specified (which is actually a block of data in itself). If B$ does not equal A$, the search goes on. If an error occurs or the tape ends, Searcher/Loader displays the error message at line 100.

If a block of data loaded corresponds to the specified filename (if B$=A$), the program jumps out of

the FOR-NEXT loop and goes to line 90. Line 90 clears the screen, erases Searcher/Loader from memory, and then loads the program that follows. When the program is loaded, the operation stops.

Although slow, Searcher/Loader does eliminate part of the hassle of cassette files.

Atari Searcher/Loader

Please refer to "COMPUTE!'s Guide to Typing In Programs" before entering this listing.

```
BI 10 REM TO SAVE A PROGRAM
      WITH A FILENAME, TYPE
      OPEN #1,8,0,"C:":?#1;"
      [FILENAME]":CLOSE #1
AO 20 DIM A$(100),B$(256)
JB 30 ? "FILENAME";:INPUT A$
BE 40 FOR I=1 TO 1.0E+97
JE 50 OPEN #1,4,0,"C:"
NO 60 TRAP 100:INPUT #1,B$
JG 70 IF B$=A$ THEN 90
JA 80 CLOSE #1:POKE 764,33:N
      EXT I
DO 90 ? "{CLEAR}":POSITION 2
      ,4:? "NEW":? :? :? "EN
      TER";CHR$(34);"C:";CHR
      $(34):? :? :? "POKE842
      ,12":POSITION 2,0:POKE
       842,13:POKE 764,33:EN
      D
PE 100 ? "BAD BLOCK. LOAD FA
       ILED. TRY AGAIN."
```

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Since programs can contain some hard-to-read (and hard-to-type) special characters, we have developed a listing system that spells out in abbreviated form the function of these control characters. You will find these special characters within curly braces. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. Commodore machines have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a new kind of special bracket. A graphics character can be listed as [<A>]. In this case, hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. Hold down SHIFT and press the space bar.

If a number precedes a symbol, such as {5 RIGHT}, {6 S}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered with the Atari logo key. Since spacing is sometimes important, any more than two spaces will be listed, for example, as: {6 SPACES}. A space is never left at the end of a line, but will be moved to the next printed line as {SPACE}. There are no special control characters found in our IBM PC/PCjr, TI-99/4A, and Apple program listings. For your convenience, we have prepared this quick-reference key for the Commodore and Atari special characters:

Atari 400/800/XL

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME		{GRN}	CTRL 6	
{HOME}	CLR/HOME		{BLU}	CTRL 7	
{UP}	SHIFT CRSR ↑		{YEL}	CTRL 8	
{DOWN}	CRSR ↓		{F1}	f1	
{LEFT}	SHIFT CRSR ←		{F2}	f2	
{RIGHT}	CRSR →		{F3}	f3	
{RVS}	CTRL 9		{F4}	f4	
{OFF}	CTRL 0		{F5}	f5	
{BLK}	CTRL 1		{F6}	f6	
{WHT}	CTRL 2		{F7}	f7	
{RED}	CTRL 3		{F8}	f8	
{CYN}	CTRL 4		←		
{PUR}	CTRL 5		↑	SHIFT ↑	

The Automatic Proofreader

Also, we have developed a simple, yet effective program that can help check your typing. Type in the appropriate Proofreader program for your machine, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader will still be active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. The IBM Proofreader is a BASIC program that lets you enter, edit, list, save, and load programs that you type. It simulates the IBM's BASIC line editor.

Using The Automatic Proofreader

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a number (on the Commodore) or a pair of letters

Now, the lowest prices ever on

3M Scotch® DISKETTES

LIFETIME WARRANTY!

$1.49 ea 5¼" SSOO Qty 50 — **$1.99** ea 5¼" OSOO Qty 50

5¼" SSOO 96TPI → **$2.29** ea 5¼" OSOO 96TPI → **$2.85** ea

SOFT SECTOR ONLY! MINIMUM ORDER 20 DISKETTES

ADD 3% FOR ORDERS UNDER 50!

FREE! FLIP N FILE 15 w/10 DISKETTES (Efl Thru 5/30/85)

These are factory-fresh 3M diskettes packed in boxes of 10 with Tyvek sleeves reinforced hubs identification labels and write protect tabs

3.5" MICRO-DISKETTES—SS-135 TPI → **$2.89** ea

LIFETIME WARRANTY ON ALL 3M SCDTCH DISKETTES!

HOURS 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRDOUCTS ANO QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD! Authorized Distributor Information Processing Products 3M

FANTASTIC LOW PRICES ON

BASF QUALIMETRIC DISKETTES!

LIFETIME WARRANTY!

$1.29 ea 5¼" SSOO Qty 20 — **$1.49** ea 5¼" OSOO Qty 20

5¼" SSOO 96TPI — **$1.45** ea 5¼" OSOO 96TPI — **$1.75** ea

PACKED IN CAROBOARD CASES!

BASF QUALIMETRIC DISKETTES have a LIFETIME WARRANTY with Tyvek sleeves reinforced hubs user identification labels and write-protect tabs

SOFT SECTOR ONLY! MINIMUM ORDER 20 DISKETTES

BASF 3.5" MICRO-FLOPPIES SSOO-135 TPI — **$2.50** ea

BASF 5¼" HIGH DENSITY FOR IBM PC-AT OSDD HO — **$4.91** ea

FOR ORDERS ONLY: **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES: **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD! Authorized Reseller Information Processing Media BASF

Incredible value!

Nashua™ Diskettes

LIFETIME WARRANTY!

$1.05 ea 5¼" SSDD Qty 50 — **$1.15** ea. 5¼" DSDD Qty 50

These are poly-bagged diskettes packaged with Tyvek sleeves reinforced hubs user identification labels and write-protect tabs NASHUA Corporation is a half-billion dollar corporation and a recognized leader in magnetic media

SOFT SECTOR ONLY! Sold in multiples of 50 only!

FOR ORDERS ONLY. **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES: **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD! Authorized Distributor NASHUA MAGNETIC MEDIA

BETTER MODEMS AT LOWER PRICES!

...and get 24-hour shipping on your DISK WORLD! orders

1200/300 Baud Avatex Modem **$189.95** ea.

300 Baud Avatex Modem **$59.95** ea

Avatex Modems have everything They re inexpensive, Hayes-compatible, Auto Dial, Auto Answer and high quality (backed by a one-year warranty)

Best of all, our combination includes a One-Year FREE subscription to MCI MAIL and special communications software for placing TOLL-FREE orders with DISK WORLD!

Orders received via MCI MAIL are shipped within 24 hours (subject to product availability)

(Cables are not included.)

FOR ORDERS ONLY **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES: **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday Friday

WE WILL BEAT ANY NATIDNALLY AOVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD! Authorized Distributor AVATEX MODEMS

DISK WORLD!

Ordering & Shipping Instructions

Shipping 5¼ & 3.5 DISKETTES—Add $3 00 per each 100 or fewer diskettes **Other Items** Add shipping charges as shown in addition to other shipping charges **Payment** VISA and MASTER-CARO accepted **CDD Orders** Add additional $3 00 Special Handling charge **APO, FPO, AK, HI & PR Orders** Include shipping charges as shown **and** additional 5% of total order amount to cover PAL and insurance **Taxes** Illinois residents only add 8% sales tax

Prices subject to change without notice
This ad supercedes all other ads
Not responsible for typographical errors
MINIMUM TOTAL QROER $35 00

FOR ORDERS ONLY: **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD!

ATHANA DISKETTES

The great unknown!

99¢ ea ← 5¼" SSOO Qty 50 — **$1.09** ea 5¼" OSDD → Qty 50

You've used these diskettes hundreds of times as copy-protected originals on some of the most popular software packages They're packed in poly-bags of 25 with Tyvek sleeves, reinforced hubs, user identification labels and write-protect tabs

LIFETIME WARRANTY!

SOFT SECTOR ONLY! Sold in multiples of 50 only

FOR ORDERS ONLY: **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD! Authorized Distributor ATHANA MAGNETIC MEDIA

DISKETTE STORAGE CASES

AMARAY MEDIA-MATE 50 A REVOLUTION IN DISKETTE STORAGE

Every once in a while someone takes the simple and makes it elegant! This unit holds 50 5¼" diskettes has grooves for easy stacking inside nipples to keep diskettes from slipping and several other features We like it! **$10.95** ea + $2.00 Shpng

DISKETTE 70 STORAGE: STILL A GREAT BUY.

Dust-free storage for 70 5¼ diskettes Six dividers included An excellent value **$11.95** + $3 00 Shpng

DISK CADDIES

The original flip up holder for 10 5¼" diskettes Beige or grey only **$1.65** ea + 20¢ Shpng

FOR ORDERS ONLY: **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES: **1-312-944-2788**

HOURS 8AM-5PM Central Time, Monday Friday

WE WILL BEAT ANY NATIDNALLY ADVERTISEO PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD! The value leader in Computer supplies And accessories.

PRINTER RIBBONS:

at extraordinary prices!

Brand new ribbons manufactured to Original Equipment Manufacturer s specifications in housings (Not re inked or spools only)

LIFETIME WARRANTY!

Epson MX-70/80	**$3.58** ea	+ 25¢ Shpng
Epson MX-100	**$4.95** ea.	+ 25¢ Shpng
Okidata Micro83	**$1.48** ea.	+ 25¢ Shpng
Okidata Micro84	**$3.66** ea	+ 25¢ Shpng.

FOR ORDERS ONLY: **1-800-621-6827** (In Illinois 1-312-944-2788)

INFORMATION & INQUIRIES **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS AND QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD!

Nail down great prices on MEMOREX diskettes!

LIFETIME WARRANTY!

$1.28 ea 5¼" SSOD Qty 20 — **$1.70** ea 5¼" OSDD Qty 20

MEMOREX DISKETTES come with heavy, lintless paper sleeves reinforced hubs write-protect tabs and user ID labels

3.5" MICRO FLOPPIES SSOO 135TPI **$2.44** ea

SOFT SECTOR ONLY! MINIMUM ORDER 20 DISKETTES

5¼" OSDD HO FOR IBM PC AT **$3.89** ea

INFORMATION & INQUIRIES: **1-312-944-2788**

HOURS 8AM-5PM Central Time Monday-Friday

WE WILL BEAT ANY NATIONALLY ADVERTISED PRICE ON THE SAME PRODUCTS ANO QUANTITIES!

DISK WORLD!, Inc.

Suite 4806 • 30 East Huron Street • Chicago Illinos 60611

DISK WORLD!

(Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari and IBM listings, the checksum is given to the left of each line number. Just type in the program, a line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore and Atari Proofreader, spaces are not counted as part of the checksum, and no check is made to see that you've typed in the characters in the right order. If characters are transposed, the checksum will still match the listing. Because of the checksum method used, do not use abbreviations, such as ? for PRINT. However, the Proofreader does catch the majority of typing errors most people make. The IBM Proofreader is even pickier; it *will* detect errors in spacing and transposition. Also, be sure you leave Caps Lock on, except when you need to enter lowercase characters.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in the cassette buffer, which is used during tape LOADs and SAVEs. Be sure to press RUN/STOP–RESTORE before you save or load a program, to get the Proofreader out of the way. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10 SPACES}"
    :FORX=1TO4:A$=A$+B$:NEXT
FORX=886TO1018:A$=A$+CHR$(PEEK(X))
    :NEXT:OPEN 1,1,1,A$:CLOSE1
```

Then press RECORD and PLAY on a blank tape, and a special version of the Proofreader will be saved to tape. Anytime you need to reload the Proofreader after it has been erased, just rewind the tape, type OPEN1:CLOSE1, then press PLAY. When READY comes back, enter SYS 886.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader will prompt you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program into the normal BASIC environment (this will replace the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE "filename",A.

VIC/64 Proofreader

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO1018:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MAOE {SPACE}AN ERROR":PRINT"IN DATA STATEMENTS.":END
120 SYS886:PRINT"{CLR}{2 OOWN}PROOFREADER ACTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 OATA 150,141,036,003,169,003
910 OATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 OATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

Atari Proofreader

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:READ A:POKE I,A:CK=CK+A:NEXT I
120 IF CK<>19072 THEN ? "Error in DATA Statements.  Check Typing.":END
130 A=USR(1536)
140 ? :? "Automatic Proofreader Now Activated."
```

```
150 END
1536 DATA 104,160,0,185,26,3
1542 DATA 201,69,240,7,200,200
1548 DATA 192,34,208,243,96,200
1554 DATA 169,74,153,26,3,200
1560 DATA 169,6,153,26,3,162
1566 DATA 0,189,0,228,157,74
1572 DATA 6,232,224,16,208,245
1578 DATA 169,93,141,78,6,169
1584 DATA 6,141,79,6,24,173
1590 DATA 4,228,105,1,141,95
1596 DATA 6,173,5,228,105,0
1602 DATA 141,96,6,169,0,133
1608 DATA 203,96,247,238,125,241
1614 DATA 93,6,244,241,115,241
1620 DATA 124,241,76,205,238,0
1626 DATA 0,0,0,0,32,62
1632 DATA 246,8,201,155,240,13
1638 DATA 201,32,240,7,72,24
1644 DATA 101,203,133,203,104,40
1650 DATA 96,72,152,72,138,72
1656 DATA 160,0,169,128,145,88
1662 DATA 200,192,40,208,249,165
1668 DATA 203,74,74,74,74,24
1674 DATA 105,161,160,3,145,88
1680 DATA 165,203,41,15,24,105
1686 DATA 161,200,145,88,169,0
1692 DATA 133,203,104,170,104,168
1698 DATA 104,40,96
```

IBM Proofreader

```
10 'Automatic Proofreader Version 2.00 (L
   ines 270,510,515,517,620,630 changed f
   rom V1.0)
100 DIM L$(500),LNUM(500):COLOR 0,7,7:KEY
     OFF:CLS:MAX=0:LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,CHR$(4)+CHR$
    (70):ON KEY(15) GOSUB 640:KEY (15) ON
    :GOTO 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PRINT"Proofre
    ader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT(LEN(L$)/W)
    -1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POKE 1052,34:P
    OKE 1054,0:POKE 1055,79:POKE 1056,13:
    POKE 1057,28:LINE INPUT L$:DEF SEG:IF
     L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$=MID$(L$,2)
    :GOTO 170
180 IF VAL(LEFT$(L$,2))=0 AND MID$(L$,3,1
    )=" " THEN L$=MID$(L$,4)
190 LNUM=VAL(L$):TEXT$=MID$(L$,LEN(STR$(L
    NUM))+1)
200 IF ASC(L$)>57 THEN 260 'no line numbe
    r, therefore command
210 IF TEXT$="" THEN GOSUB 540:IF LNUM=LN
    UM(P) THEN GOSUB 560:GOTO 150 ELSE 15
    0
220 CKSUM=0:FOR I=1 TO LEN(L$):CKSUM=(CKS
    UM+ASC(MID$(L$,I))*I) AND 255:NEXT:LO
    CATE Y,1:PRINT CHR$(65+CKSUM/16)+CHR$
    (65+(CKSUM AND 15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM THEN L$(P)=
    TEXT$:GOTO 150 'replace line
240 GOSUB 580:GOTO 150 'insert the line
260 TEXT$="":FOR I=1 TO LEN(L$):A=ASC(MID
    $(L$,I)):TEXT$=TEXT$+CHR$(A+32*(A>96
    AND A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "):COMMAND$=T
    EXT$:ARG$="":IF DELIMITER THEN COMMAN
    D$=LEFT$(TEXT$,DELIMITER-1):ARG$=MID$
    (TEXT$,DELIMITER+1) ELSE DELIMITER=IN
    STR(TEXT$,CHR$(34)):IF DELIMITER THEN
     COMMAND$=LEFT$(TEXT$,DELIMITER-1):AR
    G$=MID$(TEXT$,DELIMITER)
280 IF COMMAND$<>"LIST" THEN 410
290 OPEN "scrn:" FOR OUTPUT AS #1
300 IF ARG$="" THEN FIRST=0:P=MAX-1:GOTO
    340
310 DELIMITER=INSTR(ARG$,"-"):IF DELIMITE
    R=0 THEN LNUM=VAL(ARG$):GOSUB 540:FIR
    ST=P:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIMITER)):LAST
    =VAL(MID$(ARG$,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRST=P:LNUM=LAS
    T:GOSUB 540:IF P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(STR$(LNUM(X)
    ),2)+" "
350 IF CKFLAG=0 THEN A$="":GOTO 370
360 CKSUM=0:A$=N$+L$(X):FOR I=1 TO LEN(A$
    ):CKSUM=(CKSUM+ASC(MID$(A$,I))*I) AND
     255:NEXT:A$=CHR$(65+CKSUM/16)+CHR$(6
    5+(CKSUM AND 15))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN OPEN "lpt1:"
     FOR OUTPUT AS #1:GOTO 300
420 IF COMMAND$="CHECK" THEN CKFLAG=1:GOT
    O 290
430 IF COMMAND$<>"SAVE" THEN 450
440 GOSUB 600:OPEN ARG$ FOR OUTPUT AS #1:
    ARG$="":GOTO 300
450 IF COMMAND$<>"LOAD" THEN 490
460 GOSUB 600:OPEN ARG$ FOR INPUT AS #1:M
    AX=0:P=0
470 WHILE NOT EOF(1):LINE INPUT #1,L$:LNU
    M(P)=VAL(L$):L$(P)=MID$(L$,LEN(STR$(V
    AL(L$)))+1):P=P+1:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INPUT "Erase p
    rogram - Are you sure";L$:IF LEFT$(L$
    ,1)="y" OR LEFT$(L$,1)="Y" THEN MAX=0
    :GOTO 130:ELSE 130
500 IF COMMAND$="BASIC" THEN COLOR 7,0,0:
    ON ERROR GOTO 0:CLS:END
510 IF COMMAND$<>"FILES" THEN 520
515 IF ARG$="" THEN ARG$="A:" ELSE SEL=1:
    GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO 130
540 P=0:WHILE LNUM>LNUM(P) AND P<MAX:P=P+
    1:WEND:RETURN
560 MAX=MAX-1:FOR X=P TO MAX:LNUM(X)=LNUM
    (X+1):L$(X)=L$(X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+1 STEP -1:LN
    UM(X)=LNUM(X-1):L$(X)=L$(X-1):NEXT:L$
    (P)=TEXT$:LNUM(P)=LNUM:RETURN
600 IF LEFT$(ARG$,1)<>CHR$(34) THEN 520 E
    LSE ARG$=MID$(ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34) THEN ARG$=
    LEFT$(ARG$,LEN(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,".")=0 THEN A
    RG$=ARG$+".BAS"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"Stopped.":RET
    URN 150
650 PRINT "Error #";ERR:RESUME 150
```

Apple MLX
Machine Language Entry Program

Tim Victor, Editorial Programmer

To make it easier to enter machine language programs into your computer without typos, COMPUTE! *is introducing its MLX entry program for the Apple II series. It's our best MLX yet. It runs on the II, II+, IIe, and IIc, and with either DOS 3.3 or ProDOS.*

A machine language (ML) program is usually listed as a long series of numbers. It's hard to keep your place and even harder to avoid making mistakes as you type in the listing, since an incorrect line looks almost identical to a correct one. To make error-free entry easier, COMPUTE! generally lists ML programs for Commodore and Atari computers in a format designed to be typed in with a utility called "MLX." The MLX program uses a checksum system to catch typing errors almost as soon as they happen.

This month, COMPUTE! introduces MLX for the Apple II series. Apple MLX checks your typing on a line-by-line basis. It won't let you enter invalid characters or let you continue if there's a mistake in a line. It won't even let you enter a line or digit out of sequence. Best of all, you don't have to know anything about machine language to enter ML programs with MLX. Apple MLX makes typing ML programs almost foolproof.

Using Apple MLX

Type in and save some copies of Apple MLX on disk (you'll want to use MLX to enter future ML programs in COMPUTE!). It doesn't matter whether you type it in on a disk formatted for DOS 3.3 or ProDOS. Programs entered with Apple MLX, however, must be saved to a disk formatted with the same operating system as Apple MLX itself.

If you have an Apple IIe or IIc, make sure that the key marked CAPS LOCK is in the down position. Type RUN. You'll be asked for the starting and ending addresses of the ML program. These values vary for each program, so they're given at the beginning of the ML program listing and in the program's accompanying article. Find them and type them in.

The next thing you'll see is a menu asking you to select a function. The first is (E)NTER DATA. If you're just starting to type in a program, pick this. Press the E key, and the program asks for the address where you want to begin entering data. Type the first number in the first line of the program listing if you're just starting, or the line number where you left off if you've already typed in part of a program. Hit the RETURN key and begin entering the data.

Once you're in Enter mode, Apple MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight bytes and a checksum. When you enter a line and hit RETURN, Apple MLX recalculates the checksum from the eight bytes and the address. If you enter more or less than nine numbers, or the checksum doesn't exactly match, Apple MLX erases the line you just entered and prompts you again for the same line.

Invalid Characters Banned

Apple MLX is fairly flexible about how you type in the numbers. You can put extra spaces between numbers or leave the spaces out entirely, compressing a line into 18 keypresses. Be careful not to put a space between two digits in the middle of a number. Apple MLX will read two single-digit numbers instead of one two-digit number (F 6 means F and 6, not F6).

You can't enter an invalid character with Apple MLX. Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), nothing happens. This safeguards against entering extraneous characters. Even better, Apple MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, Apple MLX will catch your mistake.

Apple MLX also checks to make sure you're typing in the right line. The address (the number to the left of the colon) is part of the checksum recalculation. If you accidentally skip a line and try to enter incorrect values, Apple MLX won't let you continue. Just make sure you enter the correct starting address; if you don't, you won't be able to enter any of the following lines. Apple MLX will stop you.

Editing Features

Apple MLX also includes some editing features. The left- and right-arrow keys allow you to back up and go forward on the line that you are entering, so you can retype data. Pressing the CONTROL (CTRL) and D keys at the same time (*delete*) removes the character under the cursor, shortening the line by one character. Pressing CTRL-I (*insert*) puts a space under the cursor and shifts the rest of the line to the right, making the line one character longer. If the cursor is at the right end of the line, neither CTRL-D nor CTRL-I has any effect.

When you've entered the entire listing (up to the ending address that you specified earlier), Apple MLX

automatically leaves Enter mode and redisplays the functions menu. If you want to leave Enter mode before then, press the RETURN key when Apple MLX prompts you with a new line address. (For instance, you may want to leave Enter mode to enter a program listing in more than one sitting; see below.)

Display Data

The second menu choice, (D)ISPLAY DATA, examines memory and shows the contents in the same format as the program listing. You can use it to check your work or to see how far you've gotten. When you press D, Apple MLX asks you for a starting address. Type in the address of the first line you want to see and hit RETURN. Apple MLX displays program lines until you press any key or until it reaches the end of the program.

Save And Load

Two more menu selections let you save programs on disk and load them back into the computer. These are (S)AVE FILE and (L)OAD FILE. When you press S or L, Apple MLX asks you for the filename. The first time you save an ML program, the name you assign will be the program's filename on the disk. If you press L and specify a filename that doesn't exist on the disk, you'll see a disk error message.

If you're not sure why a disk error has occurred, check the drive. Make sure there's a formatted disk in the drive and that it was formatted by the same operating system you're using for Apple MLX (ProDOS or DOS 3.3). If you're trying to save a file and see an error message, the disk might be full. Either save the file on another disk or quit Apple MLX (by pressing the Q key), delete an old file or two, then run Apple MLX again. Your typing should still be safe in memory.

Apple MLX: Machine Language Entry Program

```
100 N = 9: HOME : NORMAL : PRIN
    T "APPLE MLX": POKE 34,2: O
    NERR  GOTO 610
110 VTAB 1: HTAB 20: PRINT "STA
    RT ADDRESS";: GOSUB 530: IF
     A = 0 THEN  PRINT  CHR$ (7
    ): GOTO 110
120 S = A
130 VTAB 2: HTAB 20: PRINT "END
     ADDRESS  ";: GOSUB 530: IF
     S >  = A OR A = 0 THEN  PR
    INT  CHR$ (7): GOTO 130
140 E = A
150 PRINT : PRINT "CHOOSE:(E)NT
    ER DATA";: HTAB 22: PRINT "
    (D)ISPLAY DATA": HTAB 8: PR
    INT "(L)OAD FILE  (S)AVE FI
    LE  (Q)UIT": PRINT
160 GET A$: FOR I = 1 TO 5: IF
    A$ < > MID$ ("EDLSQ",I,1) T
    HEN  NEXT : GOTO 160
170 ON I GOTO 270,220,180,200:
    POKE 34,0: END

180 INPUT "FILENAME: ";A$: IF A
    $ <  > "" THEN  PRINT  CHR$
     (4);"BLOAD";A$;",A";S
190 GOTO 150

200 INPUT "FILENAME: ";A$: IF A
    $ <  > "" THEN  PRINT  CHR$
     (4);"BSAVE";A$;",A";S;",L"
    ;E - S
210 GOTO 150
220 GOSUB 590: IF B = 0 THEN 15
    0
230 FOR B = B TO E STEP 8:L = 4
    :A = 8: GOSUB 580: PRINT A$
    ;": ";:L = 2

240 FOR F = 0 TO 7:V(F + 1) = P
    EEK (B + F): NEXT : GOSUB 5
    60:V(9) = C
250 FOR F = 1 TO N:A = V(F): GO
    SUB 5B0: PRINT A$" ";: NEXT
     : PRINT : IF  PEEK (49152)
     < 12B THEN  NEXT
260 POKE 4916B,0: GOTO 150
270 GOSUB 590: IF B = 0 THEN 15
    0
280 FOR B = B TO E STEP B

290 HTAB 1:A = B:L = 4: GOSUB 5
    B0: PRINT A$;": ";: CALL 64
    66B:A$ = "":P = 0: GOSUB 33
    0: IF L = 0 THEN 150
300 GOSUB 470: IF F < > N THEN
    PRINT CHR$ (7);: GOTO 290

310 IF N = 9 THEN GOSUB 560: IF
     C < > V(9) THEN PRINT CHR$
     (7);: GOTO 290
320 FOR F = 1 TO B: POKE B + F
    - 1,V(F): NEXT : PRINT : NE
    XT : GOTO 150
330 IF LEN (A$) = 33 THEN A$ =
    O$:P = O: PRINT CHR$ (7);
340 L = LEN (A$):O$ = A$:O = P:
    L$ = "": IF P > 0 THEN L$ =
      LEFT$ (A$,P)
350 R$ = "": IF P < L - 1 THEN
    R$ =  RIGHT$ (A$,L - P - 1)
360 HTAB 7: PRINT L$;: FLASH :
    IF P < L THEN PRINT MID$ (A
    $,P + 1,1);: NORMAL : PRINT
     R$;
370 PRINT " ";: NORMAL
380 K = PEEK (49152): IF K < 12
    8 THEN 380
390 POKE 4916B,0:K = K - 128
400 IF K = 13 THEN HTAB 7: PRIN
    T A$;" ";: RETURN
410 IF K = 32 OR K > 47 AND K <
     5B OR K > 64 AND K < 71 TH
    EN A$ = L$ + CHR$ (K) + R$:
    P = P + 1
420 IF K = 4 THEN A$ = L$ + R$
430 IF K = 9 THEN A$ = L$ + " "
     +  MID$ (A$,P + 1,1) + R$
440 IF K = 8 THEN P = P - (P >
    0)
450 IF K = 21 THEN P = P + (P <
    L)
460 GOTO 330
470 F = 1:O = 0: FOR P = 1 TO L
    EN (A$):C$ = MID$ (A$,P,1):
     IF F > N AND C$ < > " " TH
    EN  RETURN
480 IF C$ < > " " THEN  GOSUB 5
    20:V(F) = J + 16 * (O = 1)
    * V(F):O = O + 1
490 IF O > 0 AND C$ = " " OR O
    = 2 THEN O = 0:F = F + 1
500 NEXT : IF O = 0 THEN F = F
    - 1
510 RETURN
520 J = ASC (C$):J = J - 48 - 7
    * (J > 64): RETURN
530 A = 0: INPUT A$:A$ = LEFT$
    (A$,4): IF LEN (A$) = 0 THE
    N RETURN
540 FOR P = 1 TO LEN (A$):C$ =
    MID$ (A$,P,1): IF C$ < "0"
    OR C$ > "9" AND C$ < "A" OR
     C$ > "Z" THEN A = 0: RETUR
    N
550 GOSUB 520:A = A * 16 + J: N
    EXT : RETURN
560 C = INT (B / 256):C = B - 2
    54 * C - 255 * (C > 127):C
    = C - 255 * (C > 255)
570 FOR F = 1 TO 8:C = C * 2 -
    255 * (C > 127) + V(F):C =
    C - 255 * (C > 255): NEXT :
     RETURN
580 I = FRE (0):A$ = "": FOR I
    = 1 TO L:T =  INT (A / 16):
    A$ =  MID$ ("0123456789ABCD
    EF",A - 16 * T + 1,1) + A$:
    A = T: NEXT : RETURN
590 PRINT "FROM ADDRESS ";: GOS
    UB 530: IF S > A OR E < A O
    R A = 0 THEN B = 0: RETURN
600 B = S + B * INT ((A - S) /
    B): RETURN
610 PRINT "DISK ERROR": GOTO 15
    0
```

SpeedScript 3.0

All Machine Language Word Processor For Apple

Charles Brannon, Program Editor
Apple Adaptation By Kevin Martin, Editorial Programmer

COMPUTE! *concludes its* SpeedScript 3.0 *series this month with a version for Apple II-series computers with DOS* 3.3 *and at least 48K RAM. Originally written for the Commodore 64 and VIC-20,* SpeedScript *has also been adapted for Atari computers (*COMPUTE!, *May 1985) and has become extremely popular. It compares favorably with commercial programs and has some features never seen before in an Apple word processor.*

SpeedScript 3.0, though compact in size (5.5K), has most of the functions you expect in a full-featured word processor. *SpeedScript* is also very easy to learn and use. You type in everything first; preview and make corrections on the screen; insert and delete words, sentences, and paragraphs; then print out an error-free draft, letting *SpeedScript* take care of things like margins, centering, headers, and footers.

The Apple version of *SpeedScript 3.0,* and all other Apple programs in this issue, may be ordered on disk directly from COMPUTE! Publications. Call **TOLL FREE 1-800-334-0868** (in NC 1-919-275-9809) to charge your order 8:30 a.m.–7:00 p.m. Eastern Time, Monday through Friday. Or send check or money order ($12.95 plus $2.00 shipping and handling) to:

COMPUTE! Publications, Inc.
P.O. Box 5058
Greensboro, NC 27403 USA

Readers outside the United States and Canada add $3.00 shipping and handling. All orders must be prepaid in US funds.

Special Typing Instructions

Apple *SpeedScript* is the longest Apple machine language program we've ever published, but COMPUTE!'s new "Apple MLX" entry system helps you type it right the first time. MLX can detect most typing errors as they happen. (See the Apple MLX article elsewhere in this issue.) MLX also lets you type *SpeedScript* in more than one sitting. Although the program listing is lengthy, we guarantee the effort will be worthwhile. If you prefer, you can order Apple *SpeedScript 3.0* (and all other Apple programs in this issue) on disk directly from COMPUTE! Publications at a nominal cost (see box).

To begin entering the data for *SpeedScript,* boot up your Apple with a DOS 3.3 startup disk in the drive. As the MLX article states, programs entered with MLX must be saved to a disk with the same operating system format as the disk from which MLX was loaded. Since this version of *SpeedScript* works only with DOS 3.3, you must load Apple MLX from a DOS 3.3 disk. If you have a IIe or IIc that came with the ProDOS operating system, you must obtain a copy of DOS 3.3 before entering *SpeedScript.*

Because the machine language data for *SpeedScript* resides in the same area of memory where BASIC programs are normally loaded, it's necessary in this case to reconfigure memory before loading MLX to enter *SpeedScript.* Otherwise, the *SpeedScript* data you enter with MLX will overwrite the MLX program itself as you type. To reconfigure memory, type the following line in direct mode (no line number) and hit RETURN:

```
POKE 104,32: POKE 8192,0: NEW
```

You must *always* enter this line before loading MLX to enter *SpeedScript* data. It is, however, *not* necessary to enter this line before loading the completed *SpeedScript* program.

Now load and run Apple MLX. Answer the first two questions that MLX asks like this:

```
STARTING ADDRESS? 0800
ENDING ADDRESS? 1E45
```

An options menu appears next. Press E to Enter the program. Now type the address at which you'd like to start typing. If you're just beginning to type the listing, you'd enter 0800. The screen then shows the first prompt, the number 0800 followed by a colon (:). Type in each two-digit number shown in the *SpeedScript* listing (some of the digits are letters, because the numbers are in hexadecimal). You don't need to type the spaces shown in the listing, but you can for the sake of readability. MLX does not let you type illegal characters.

The last number you enter in each line is a *checksum.* If you type the line correctly, the checksum calculated by MLX should match the checksum number you typed in. If it doesn't match, MLX makes you retype the line. MLX is not foolproof, though. It's quite rare, but it's possible that an error in one number could be offset by an error in another. MLX will help catch your errors, but you still must be careful.

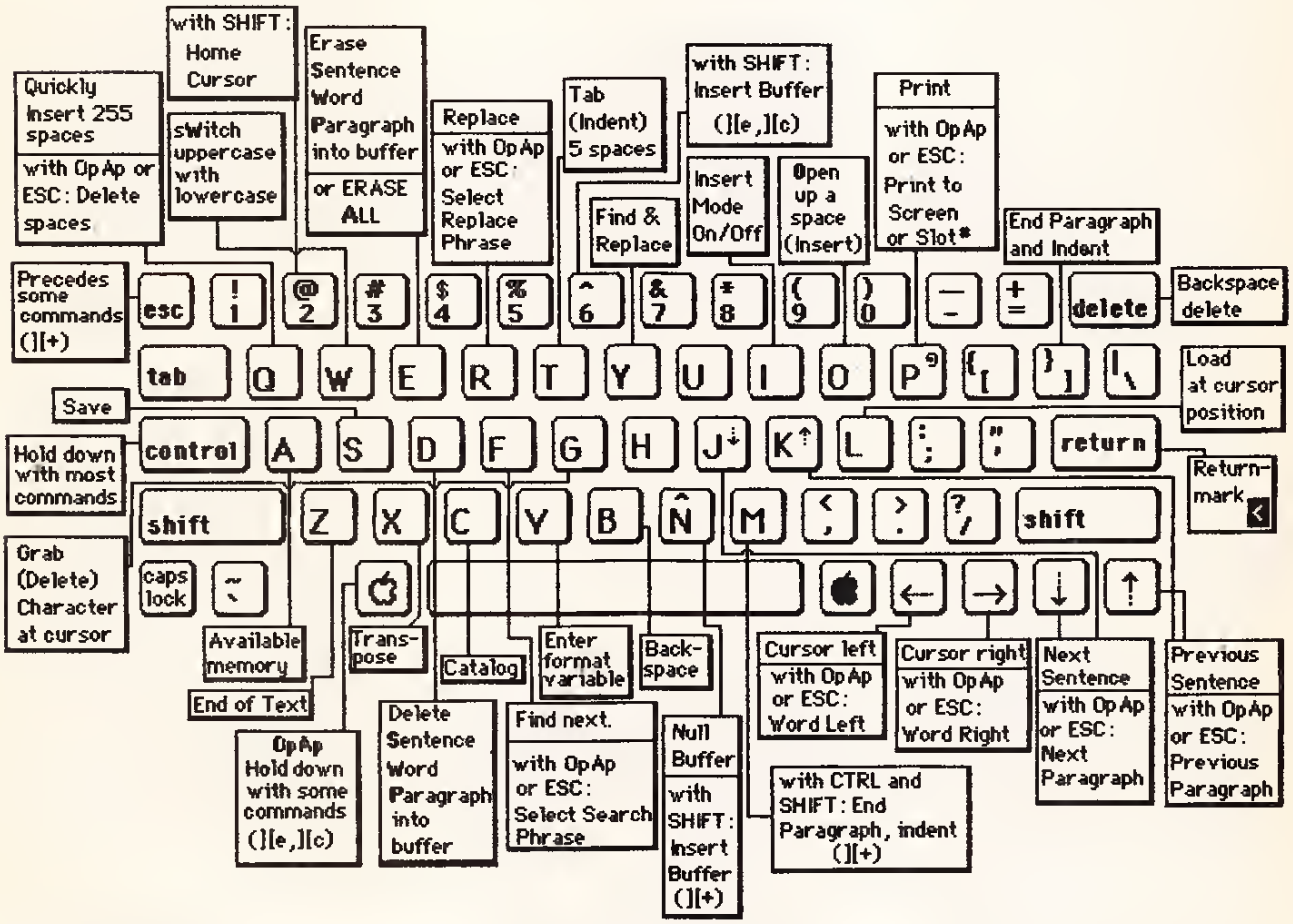
Typing In Multiple Sittings

If you want to stop typing the listing at some point and pick up later, press RETURN at the address prompt without typing anything.

Apple SpeedScript 3.0 Keyboard Map

Use CTRL or control with most commands

Apple IIc Keyboard Shown. Apple IIe,II+ keyboard similar



Write down the address number you stopped at. The options menu reappears, and you can save your typing at this point. To continue entering data the next time, boot your system, enter the direct-mode line given above to configure memory, and load MLX. Answer the STARTING ADDRESS? and ENDING ADDRESS? questions with the same values you used the first time, 0800 and 1E45. Then select Load from the options menu, then press E to enter data. Give MLX the address number you previously stopped at. Now continue typing as before.

When you finish all typing, MLX returns you to the options menu, where you can save the finished program.

MLX has now created a binary file on disk. To run it, reboot the machine to clear out memory, then from BASIC type BRUN *filename*, where *filename* is the name you specified when saving *SpeedScript* on disk with MLX. *SpeedScript* automatically loads and runs. If you prefer, you can write a short program:

```
10 PRINT CHR$(4);"BRUN filename"
```

and save it as the HELLO file on the disk (INIT HELLO to format and set up a blank disk). This makes *SpeedScript* load and run automatically when you boot up.

Do You Have Lowercase?

When you BRUN *SpeedScript*, you'll first be asked, LOWERCASE? [Y/N]. If you are running *SpeedScript* on an Apple IIe or IIc, or have a lowercase character generator in your Apple II+, press Y for Yes. Otherwise, press N. This adjusts *SpeedScript* for your machine.

After you've answered the prompt, you'll see a blank 40-column screen with a blinking underline cursor. The first line on the screen is in inverse video, white with black letters. *SpeedScript* presents all messages on this *command line*. The remaining 23 lines of the screen are used to enter, edit, and

display your document. You cannot enter text in 80 columns, although you can print the document to the screen in 80 columns, if you have the appropriate hardware.

The cursor shows where the next character you type will appear on the screen. *SpeedScript* lets you move the cursor anywhere within your document, making it easy to find and correct errors.

Entering Text

To begin using *SpeedScript*, just start typing. If you have an Apple II+ without a lowercase character generator and SHIFT key modification, you'll need to follow a special procedure because the Apple II+ SHIFT key does not work with alphabetic characters. (Lowercase adapters and SHIFT key modifications are available for the Apple II+; see your Apple dealer.)

For lowercase text, just type normally. On an Apple II+ without a lowercase adapter, lowercase text appears on the screen as uppercase. When you need to specify an uppercase letter, press the ESC key before typing that letter. An uppercase letter appears on the screen in inverse video (with the colors switched). The next character you type will appear as a normal uppercase character, representing lowercase.

This is the convention used by most Apple word processors when lowercase is not available. While this may seem awkward, it overcomes the uppercase-only limitation of the Apple II+ and becomes second nature after a while. For example, to enter:

Who won the World Series?

you'd type:

```
ESC WHO WON THE ESC WORLD
ESC SERIES?
```

which appears onscreen as:

```
WHO WON THE WORLD SERIES?
```

With an Apple II+ with the SHIFT key modification or an Apple IIe/IIc, you type as you would on a typewriter, holding down the SHIFT key while typing the uppercase letter. Be sure to disengage the CAPS LOCK key if you want to type lowercase.

When the cursor reaches the right edge of the screen, it automatically jumps to the beginning of the next line, just as in BASIC. But unlike BASIC, *SpeedScript* never splits words at the right edge of the screen. If a word you're typing won't fit at the end of one line, it's instantly moved to the next line. This feature, called *word-wrap* or *parsing*, also helps make your text more readable.

Scrolling And Screen Formatting

When you finish typing on the last screen line, *SpeedScript* automatically scrolls the text upward to make room for a new line at the bottom. Imagine the screen as a 23-line window on a long, continuous document. There's room in memory for 27904 characters on a 48K machine, or about 10–15 pages of text. (Unfortunately, *SpeedScript 3.0* cannot make use of the extra memory available with 64K or 128K.) To check at any time how much space is available, press **CTRL-A** (hold down the CTRL key while pressing the A key). The number appearing in the command line indicates how much *available* room remains for characters of text.

If you're used to a typewriter, you'll have to unlearn some habits if this is your first experience with word processing. Since the screen is only 40 columns wide, and most printers have 80-column carriages, it doesn't make sense to press RETURN at the end of each line as you do on a typewriter. *SpeedScript*'s word-wrap takes care of this automatically. Press RETURN only when you want to force a carriage return to end a paragraph or limit the length of a line. A *return-mark* appears on the screen as an inverse less-than sign (<).

Using The Keyboard

Most features are accessed with control-key commands—you hold down CTRL while pressing another key. In this article, control-key commands are abbreviated **CTRL-*x*** (where *x* is the key you press in combination with CTRL). An example is the CTRL-A mentioned above to check on available memory. CTRL-E means hold down CTRL and press E.

Some commands have special options. On the Apple II+, you'll sometimes need to press ESC before the CTRL key, as in ESC-CTRL-E. You first press ESC, then release it and press CTRL and E together. If your Apple II+ has the SHIFT key modification, you can press SHIFT instead of ESC, but you must press it simultaneously with CTRL and the command key.

On the Apple IIe and IIc, you hold down the Open Apple key (the key with a hollow Apple symbol) while pressing the CTRL key combination. This is represented in this article as **OpAp-CTRL-E**. Other keys are referenced by name or function, such as DELETE for the backspace key, *carat* for the carat (^) symbol (SHIFT-N on the Apple II+ or SHIFT-6 on the Apple IIe/IIc), or *cursor-left* for the ← key. See the figure for a complete quick-reference chart of all keyboard commands.

The Apple II+ keyboard does not support all the keys used by *SpeedScript*, such as cursor-up and cursor-down (↑ ↓), but these commands can still be accessed with CTRL-key combinations. Because *SpeedScript* uses almost every key, not all combinations are especially mnemonic. Most keys, though, stand for the name of the function they perform.

Some keys let you move the cursor to different places in the document to make corrections or scroll text into view. You can move the cursor by character, word, sentence, or paragraph. Here's how to control the cursor:

- The **cursor-left/right** keys (←/→) move the cursor a single space in either direction. By preceding this key with ESC on the Apple II+, or by holding down the Open Apple key while pressing the key on the Apple IIe/IIc, you can move the cursor to the beginning of the next (→) or previous (←) word.

- The **cursor-up/down** keys (↑/↓ on the IIe/IIc, **CTRL-J/CTRL-K** on the II+) move the cursor to the beginning of either the previous or next sentence. On the IIe/IIc, hold down the Open Apple key as you press the arrow to move to the beginning of the next (↓) or previous (↑) paragraph. On the Apple II+, press CTRL-K to move the cursor to the beginning of the next sentence, and press CTRL-J to move the cursor to the beginning of the previous sentence. Press ESC-CTRL-K to move

the cursor to the beginning of the next paragraph, or ESC-CTRL-J to move the cursor to the beginning of the previous paragraph. (A paragraph is defined as any sequence of characters ending in a return-mark.)

• Pressing **CTRL-@** (CTRL-SHIFT-2 on the IIe/IIc, CTRL-SHIFT-P on the II+), puts the cursor *at* the top of the screen. If the cursor is already at the top of the screen, CTRL-@ moves the cursor to the top of the document. So to quickly move to the beginning of the document, press CTRL-@ twice.

• **CTRL-Z** moves the cursor to the end of the document, scrolling if necessary. It's easy to remember since Z is at the *end* of the alphabet.

Making Corrections

Sometimes you'll have to insert some characters to make a correction. Use **CTRL-O** to *open up* a single space. Merely position the cursor at the point where you want to insert a space, and press CTRL-O.

It can be tedious to use CTRL-O to open up enough space for a whole sentence or paragraph. For convenience, *SpeedScript* has an insert mode that automatically inserts space for each character you type. In this mode, you can't type over characters; everything is inserted at the cursor position. To enter insert mode, press **CTRL-I.** To cancel insert mode, press CTRL-I again. To let you know you're in insert mode, the cursor changes from a blinking underline to a blinking inverse underline (which looks like a solid square). The cursor changes back to a blinking underline when you exit insert mode. Because of keyboard redundancy, the TAB key on the Apple IIe/IIc works just like CTRL-I.

Insert mode is the easiest way to insert text, but it can become too slow when inserting near the top of a very long document because it must move all the text following the cursor position. So *SpeedScript* has even more ways to insert blocks of text.

One way is to use **CTRL-T** (*tab*). It is programmed in *SpeedScript* to act as a five-space margin indent. To end a paragraph and start another, press RETURN twice and press CTRL-T. A shortcut for this is **CTRL-]** on the Apple IIe/IIc and **CTRL-SHIFT-M** on the Apple II+; these keystrokes automatically insert two return-marks and indent the margin. CTRL-T always inserts; you don't need to be in insert mode. You can also use CTRL-T to open up more space than CTRL-O. (You cannot set or clear tab stops in *SpeedScript* as you can with some word processors.) No matter how much space you want to insert, each insertion takes the same amount of time. So CTRL-T can insert five spaces five times faster than pressing CTRL-O five times.

There's an even better way, though. Press **CTRL-Q** to *quickly* insert 255 spaces (it does not insert a line; use RETURN for that). You can press it several times to open up as much space as you need. And CTRL-Q is quick indeed. It inserts 255 spaces as fast as CTRL-O opens up one space. Now just type the text you wanted to insert over the blank space. (You don't want to be in CTRL-I insert mode when you use this trick; that would defeat its purpose.)

Since DELETE (backspace) is also slow when working with large documents (it, too, must move all text following the cursor), you may prefer to use the cursor-left key to backspace when using this method.

After you're done inserting, there may be some inserted spaces left over that you didn't use. Just press **ESC-CTRL-Q** on the II+ or **OpAp-CTRL-Q** on the IIe/IIc. This instantly deletes all extra spaces between the cursor and the start of following text. It's also useful whenever you need to delete a block of spaces for some reason.

Erasing Text

To erase the character on which the cursor is sitting, press **CTRL-G** (to *grab* the character). The character highlighted by the cursor is removed, and all following text is moved toward the cursor to fill the empty space.

Press **CTRL-B** (*backspace*) on the II+ or the **DELETE** key on the IIe/IIc to delete the character to the left of the cursor. All the following text is moved with the cursor to fill the empty space.

These keys are fine for minor deletions, but it could take all day to delete a whole paragraph this way. So *SpeedScript* has two commands that can delete an entire word, sentence, or paragraph at a time. **CTRL-E** erases text *after* (to the right of) the cursor position (and can also erase all text), and **CTRL-D** deletes text *behind* (to the left of) the cursor.

To use the CTRL-E erase mode, first place the cursor at the beginning of the word, sentence, or paragraph you want to erase. Then press CTRL-E. The command line shows the message "ERASE (S,W,P,A): RETURN TO EXIT." Press S to erase a sentence, W for a word, or P for a paragraph. Each time you press one of these letters, the text is quickly erased. You can keep pressing S, W, or P until you've erased all the text you wish. Then press RETURN to exit the erase mode.

You also use CTRL-E to erase all text from memory when you want to start a new document. To erase all text, press CTRL-E, then press the A (*all*) key. A prompt appears: ERASE ALL TEXT: ARE YOU SURE (Y/N). Press Y to perform the irreversible deed. You cannot recover any text erased this way. Press N or any other key to cancel this function.

The CTRL-D delete mode works similar to CTRL-E, but deletes only one word, sentence, or paragraph at a time. First, place the cursor after the word, sentence, or paragraph you want to delete. Then press CTRL-D. Next, press S, W, or P for sentence, word, or paragraph. The text is immediately deleted and you return to editing. You don't need to press RETURN to exit the CTRL-D delete mode unless you pressed this key by mistake. *(In general, you can escape from any function in SpeedScript by simply pressing RETURN.)* CTRL-D is most convenient when the cursor is already past what you've been typing.

The Text Buffer

When you erase or delete with CTRL-E and CTRL-D, the text isn't lost forever (unless you've performed an Erase All). *SpeedScript* remembers what you've removed by storing deletions in a separate area of memory called a *buffer*. The buffer is a failsafe device. If you erase too much, or change your mind, just press **CTRL-*carat*** (CTRL-SHIFT-6 on the IIe/IIc, CTRL-SHIFT-N on the II+) to restore the deletion. However, be aware that *SpeedScript* remembers only the last erase or

delete you performed. It's not to hard to remember this command, since the carat is used in paper-and-pencil editing to signify an insertion. Be sure you don't press CTRL-N without SHIFT, since CTRL-N is the command to clear out (*null*) the buffer.

Another, more powerful, use of this buffer is to move or copy sections of text. To move some text from one location in your document to another, first erase or delete it with CTRL-E or CTRL-D. Then move the cursor to where you want the text to appear and press CTRL-*carat*. CTRL-*carat* instantly inserts the contents of the buffer at the cursor position. If you want to copy some text from one part of your document to another, just erase or delete it with CTRL-E or CTRL-D, restore it at the original position with CTRL-*carat*, then move the cursor elsewhere and press CTRL-*carat* to restore it again. You can retrieve the buffer with CTRL-*carat* as many times as you like. If there is no room left in memory for inserting the buffer, you'll see the message NO ROOM.

Important: The CTRL-E erase mode lets you erase up to the maximum size of the buffer (2K), and CTRL-E also removes the previous contents of the buffer. The buffer is also erased with the ERASE ALL option of CTRL-E. Keep this in mind if there's something in the buffer you'd rather keep. If you don't want the buffer to be erased, hold down the Open Apple key (or precede with ESC on the II+) when you press CTRL-E. This preserves the buffer contents and adds newly erased text to the buffer.

If you ever need to erase the contents of the buffer, press **CTRL-N** (*null buffer*).

Search And Replace

SpeedScript has a Find command that searches through your document to find a selected word or phrase. A Replace option lets you automatically change one word to another throughout the document.

ESC-CTRL-F or **OpAp-CTRL-F** (*find*) lets you define a search phrase, **ESC-CTRL-R** or **OpAp-CTRL-R** (*replace*) lets you define a replace phrase, and **CTRL-Y** is for automatically searching and replacing.

Searching is a two-step process. First you need to tell *SpeedScript* what to search for, then you trigger the actual search. Hold down Open Apple and press CTRL-F (on the Apple II+, press ESC, then CTRL-F). The command line prompts FIND:. Type in what you'd like to search for, the *search phrase*. If you press RETURN alone without typing anything, the Find command is canceled.

When you are ready to search, press **CTRL-F**. *SpeedScript* looks for the next occurrence of the search phrase *starting from the current cursor position*. If you want to hunt through the entire document, press CTRL-@ twice to move the cursor to the very top before beginning the search. Each time you press CTRL-F, *SpeedScript* looks for the next occurrence of the search phrase and places the cursor at the start of the phrase. If the search fails, you'll see the message NOT FOUND.

CTRL-R works together with CTRL-F. After you've specified the search phrase with OpAp-CTRL-F or ESC-CTRL-F, press OpAp-CTRL-R or ESC-CTRL-R to select the replace phrase. (You can press RETURN alone at the REPLACE WITH: prompt to select a *null* replace phrase. When you hunt and replace, this deletes the located phrase.) To manually search and replace, start by pressing CTRL-F. After *SpeedScript* finds the search phrase, press CTRL-R if you want to replace the phrase. If you don't want to replace the phrase, don't press CTRL-R. You are not in a special search and replace mode. You're free to continue writing at any time.

CTRL-Y links CTRL-F and CTRL-R together (think of the two branches of the Y linking together Find and Replace). It first asks FIND:, then REPLACE:, then automatically searches and replaces throughout the document starting at the cursor position.

There are a few things to watch out for when using search and replace. First, realize that if you search for "the," *SpeedScript* finds the embedded "the" in words like "therefore" and "heathen." If you changed all occurrences of "the" to "cow," these words would become "cowrefore" and "heacown." If you want to find a single word, include a space as the first character of the word, since almost all words are preceded by a space. Naturally, if you are replacing, you need to include the space in the replace phrase, too.

Also, *SpeedScript* distinguishes between upper- and lowercase. The word "Meldids" does not match with "meldids." *SpeedScript* will not find a capitalized word unless you capitalize it in the search phrase. To cover all bases, you will sometimes need to make two passes at replacing a word. Keep these things in mind when using CTRL-Y, since you don't have a chance to stop a linked find and replace.

Storing Your Document

To store your text, press **CTRL-S**. You'll see the prompt SAVE:. Type in the filename and press RETURN. If you need to access a second disk drive, precede the filename with **2:**. This becomes the default drive for future disk access. To return to drive 1, precede the filename with **1:**. If the filename you specify coincides with one already on the disk, the existing file will be overwritten by the new one.

CTRL-S always saves the entire document. The cursor position within the document is not important.

When the SAVE is complete, *SpeedScript* reports NO ERRORS if all is well, or gives a message like DISK FULL if not. Check your DOS or BASIC manual for a list of error messages and their causes.

Press CTRL-C to display the disk catalog. The catalog pauses when the screen is full, waiting for you to press a key to continue. When the catalog is finished, press RETURN to return to editing.

Loading A Document

To recall a previously saved document, press CTRL-L. Answer the LOAD: prompt with the filename. Again, you can precede the filename with **1:** or **2:** to switch drives. *SpeedScript* loads the file and should display NO ERRORS. Otherwise, *SpeedScript* reports the error.

The position of the cursor is important before loading a file. Documents start loading at the cursor position, so be sure to press CTRL-@ twice or CTRL-E and A (Erase All) to move the cursor to the start of text, unless you want to merge two documents. When you press CTRL-L to

load, a flashing asterisk appears to warn you if the cursor is not at the top of the document.

To merge two or more files, simply load the first file, press CTRL-Z to move the cursor to the end of the document, and then load the file you want to merge. Do not place the cursor somewhere in the middle of your document before loading. A Load does not insert the text from disk, but overwrites all text after the cursor position. The last character loaded becomes the new end-of-text pointer, and you cannot access any text that appears ahead of this pointer.

Since *SpeedScript* stores documents as binary files, you cannot read a text file in BASIC, load a BASIC program into *SpeedScript*, or upload a text file with a modem. However, Program 2 is a file conversion program which allows these functions. It converts *SpeedScript* binary files into ASCII text files. It can also convert an ASCII text file into a *SpeedScript* binary file. This lets you convert word processing files from other word processors, or change a *SpeedScript* document into a text file suitable for uploading. You can even edit BASIC programs with *SpeedScript*. Add the following line to your BASIC program and run it. It creates a text file on disk of your BASIC listing.

```
0 PRINT CHR$(4);"OPEN filename"
  :PRINT CHR$(4)"WRITE
  filename":LIST
```

Run Program 2 to convert the BASIC text file to a *SpeedScript* file. You can then load this file into *SpeedScript* for editing. Save this file back to disk, run Program 2 to convert it back to a text file, then in BASIC use EXEC *filename* to read the text file back into BASIC. Although this seems rather tedious, you may find it quite worthwhile when working with long programs. A similar technique can be used to edit files written by other applications.

Note: Delete any *SpeedScript* print formatting commands (described below) before converting a text file to an ASCII file. Otherwise, they will not be converted correctly.

Additional Features

SpeedScript has a few commands that don't do much, but are nice to have. **CTRL-X** exchanges the character under the cursor with the character to the right of the cursor. Thus you can fix transposition errors with a single keystroke. **CTRL-W** (think *sWitch)* changes the character under the cursor from uppercase to lowercase or vice versa.

Apple *SpeedScript* traps the RESET key. RESET or CTRL-RESET always returns you to editing mode. There is no way to exit *SpeedScript* once you've run it, short of rebooting.

PRINT!

If you already think *SpeedScript* has plenty of commands, wait until you see what the printing package offers. *SpeedScript* supports an array of powerful formatting features. It automatically fits your text between left and right margins you can specify. You can center a line or block it against the right margin. *SpeedScript* skips over the perforation on continuous-form paper, or can wait for you to insert single-sheet paper. A line of text can be printed at the top of each page (a *header*) and/or at the bottom of each page (a *footer*), and can include automatic page numbering, starting with whatever number you like.

SpeedScript can print on different lengths and widths of paper, and single-, double-, triple-, or any-spacing is easy. You can print a document as big as can fit on a disk by linking several files together during printing. You can print to the screen or to a file instead of to a printer. Other features let you send special codes to the printer to control features like underlining, boldfacing, italics, and double-width type (depending on the printer).

But with all this power comes the need to learn additional commands. Fortunately, *SpeedScript* sets most of these variables to a *default* state. If you don't change these settings, *SpeedScript* assumes a left margin of five, a right margin position of 75, no header or footer, single-spacing, and continuous paper page feeding. You can change these default settings if you want (see below).

Before printing, be sure the paper in your printer is adjusted to top-of-form (move the paper perforation just above the printing element). One additional note: Some printers incorporate an automatic skip-over-perforation feature. The printer skips to the next page when it reaches the bottom of a page. Since *SpeedScript* already controls paper feeding, you need to turn off this automatic skip-over-perf feature before running *SpeedScript*, or paging won't work properly.

To begin printing, press **CTRL-P.** If your printer is attached, powered on, and selected (online), *SpeedScript* begins printing immediately. To cancel printing, press CTRL and the RESET key.

If you need to print to an RS-232 printer or to a printer in a slot other than slot #1, press **ESC-CTRL-P** (Apple II+) or **OpAp-CTRL-P** (Apple IIe/IIc). This brings up the prompt PRINT TO: SCREEN, PRINTER? Press S to print to the screen. If you don't have lowercase, the screen display won't make much sense, although you can still see where pages break. If you have an Apple IIc, an Apple IIe with the 80-column card, or an Apple II+ with a compatible 80-column card, *SpeedScript* automatically prints to the screen in 80 columns, simulating the printer.

If you select P, you'll be asked for the slot number. Output is sent to the slot number you select. It's similar to PR# in BASIC. If you print to slot 6 (the disk drive), you'll cause the system to reboot, so be careful.

Formatting Commands

The print formatting commands are single letters embedded in text, such as L for left margin. To enter a formatting variable, press **CTRL-V.** You'll see the prompt ENTER FORMAT VARIABLE:. Now press any key. The print formatting commands must be distinguished from normal text, so they appear on-screen in flashing inverse video with the text and background colors switched. All lettered printer commands should be entered without the SHIFT key. During printing, *SpeedScript* treats these characters as printing commands.

There are two kinds of printing commands, which we'll call Stage 1 and Stage 2. Stage 1 commands usually control variables such as left margin and right margin. Most are followed by a number, with no space between the command and the number. Stage 1 commands are

executed before a line is printed.

Stage 2 commands, like centering and underlining, are executed while the line is being printed. Usually Stage 1 commands must be on a line of their own, although you can group several Stage 1 commands together on a line. Stage 2 commands are by nature embedded within a line of text. Again, remember to press CTRL-V to enter the boldface characters shown below.

Stage 1 Commands

L Left margin. Follow with a number from 0 to 255. Use 0 for no margin. Defaults to 5.

R Right margin position, a number from 1 to 255. Defaults to 75. Be sure the right margin value is greater than the left margin value, or *SpeedScript* will go bonkers.

T Top margin. The position at which the first line of text is printed, relative to the top of the page. Defaults to 5. The header (if any) is always printed on the first line of the page, before the first line of text.

B Bottom margin. The line at which printing stops before continuing to the next page. Standard 8½ × 11-inch paper has 66 lines. Bottom margin defaults to the fifty-eighth line. Don't make the bottom margin greater than the page length.

P Page length. Defaults to 66. If your printer does not print six lines per inch, multiply lines-per-inch by 11 to get the page length. European paper is usually longer than American paper—11⅝ or 12 inches. Try a page length of 69 or 72.

S Spacing. Defaults to single-spacing. Follow with a number from 1 to 255. Use 1 for single-spacing, 2 for double-spacing, 3 for triple-spacing.

@ Start numbering *at* page number given. Page numbering normally starts with 1.

? Disables printing until selected page number is reached. For example, a value of 3 would start printing the third page of your document. Normally, *SpeedScript* prints starting with the first page.

X Sets the page width, in columns (think *a cross*). Defaults to 80. You need to change this for the sake of the centering command if you are printing in double-width or condensed type, or are using a 40-column or wide-carriage printer.

N Forced paging. Normally, *SpeedScript* prints the footer and moves on to the next page only when it has finished a page, but you can force it to continue to the next page by issuing this command. It requires no numbers.

M Margin release. Disables the left margin for the next printed line. Remember that this executes before the line is printed. It's used for outdenting.

W Page wait. This command should be placed at the beginning of your document before any text. With page wait turned on, *SpeedScript* prompts you to INSERT NEXT SHEET, PRESS RETURN when each page is finished printing. Insert the next sheet, line it up with the printhead, then press RETURN to continue. Page wait is ignored during disk or screen output.

J Select automatic linefeeds after carriage return. Like **W**, this command must be placed before any text. Don't use this command to achieve double-spacing, but only if all text prints on the same line with some printers.

I Information. This works like REM in BASIC. You follow the command with a line of text, up to 255 characters, ending in a return-mark. This line will be ignored during printing, and is handy for making notes to yourself such as the filename of the document.

H Header define and enable. The header must be a single line of text (up to 254 characters) ending in a return-mark. The header prints on the first line of each page. You can include Stage 2 commands such as centering and page numbering in a header. You can use a header by itself without a footer. The header and footer should be defined at the top of your document, before any text. If you want to prevent the header from printing on the first page, put a return-mark by itself at the top of your document before the header definition.

F Footer define and enable. The footer must be a single line of text (up to 254 characters) ending in a return-mark. The footer prints two lines prior to the last line of each page. As with the header, you can include Stage 2 printing commands, and you don't need to set the header to use a footer.

G Go to (link) next file. Put this command as the last line in your document. Follow the command with the filename (with no spaces between the **G** and the filename), including the drive number prefix **1:** or **2:**, if appropriate. After the text in memory is printed, the link command loads the next file into memory. You can continue linking successive files, but don't include a link in the last file. Before you start printing a linked file, make sure the first of the linked files is in memory. When printing is finished, the last file linked to will be in memory.

Stage 2 Commands

These commands either precede a line of text or are embedded within one.

C Centering. Put this at the beginning of a line you want to center. This centers only one line ending in a return-mark. Repeat this command at the beginning of every line you want centered. Centering uses the page-width setting (see above) to properly center the line. To center a double-width line, either set the page width to 40 or pad out the rest of the line with an equal number of spaces. If you use double width, remember that the spaces preceding the centered text will be double-wide spaces.

When *SpeedScript* encounters this command, it prints the current page number. You usually embed this within a header or footer.

U A simple form of underlining. It works only on printers that recognize CHR$(8) as a backspace and CHR$(95) as an underline character. Underlining works on spaces, too. Use the first **U** to start underlining, and another one to turn off underlining.

Fonts And Styles

Most dot-matrix printers are capable of more than just printing text at ten characters per inch. Some printers have several character sets, with italics and foreign language characters. Most can print in double width (40 characters per line), condensed (132 characters per line), and in either pica or elite. Other features may include programmable characters, programmable tab stops, and graphics modes. Many word processors customize themselves to a particular printer, but for flexibility

SpeedScript was purposely designed not to be printer-specific. Instead, *SpeedScript* lets you define your own Stage 2 printing commands.

You define a programmable *printkey* by choosing any character that is not already used for other printer commands. The numbers 0-9, most symbols, and some alphabetic characters are available for printkeys. You enter these commands like printer commands with CTRL-V. The printkeys are like variables in BASIC.

To define a printkey, press CTRL-V, then type the key you want to assign as the printkey, then an equals sign (=), and finally the ASCII value to be substituted for the printkey during printing. Now whenever *SpeedScript* encounters the printkey embedded in text, it prints the character with the ASCII value you previously defined. (If you have trouble passing some printkeys to the printer, try adding 128 to the ASCII value you wish to send. Otherwise, some printer interfaces intercept the printkey.)

For example, to define the + key as the letter z, you first look up the ASCII value of the letter z (in either your printer manual or in the BASIC manual). The ASCII value of the letter z is 122, so the definition is:

```
+=122<
```

Now, anywhere you want to print the letter z, substitute the printkey:

```
Gad+ooks! The +oo is +any!<
```

This would appear on paper as:

Gadzooks! The zoo is zany!

More practically, here's how you could program italics on an Epson MX-80 compatible printer. You switch on italics by sending an ESC (a character with an ASCII value of 27), then the character 4. You turn off italics by sending ESC 5. So define * as the escape code. Anywhere you want to print a word in italics, bracket it with *4 and *5.

You can similarly define whatever codes your printer uses for features like double width or emphasized mode. For your convenience, four of the printkeys are predefined, though you can change them. The keys 1-4 are defined as 27, 14, 15, and 18, common values for most printers. On most printers, CHR$(27) is the ESCape key, CHR$(14) starts double-width printing, CHR$(15) either stops double width or starts condensed characters, and CHR$(18) usually cancels condensed characters.

Keep one thing in mind about printkeys. *SpeedScript* always assumes it is printing to a rather dumb, featureless printer, the least common denominator. *SpeedScript* doesn't understand the intent of a printkey; it justs sends out its value. So if you make one word within a line double width, it may make the line overflow the specified right margin. There's no way for *SpeedScript* to include built-in font and typestyle codes without being customized for a particular printer, since no set of codes is universal to all printers.

Hints And Tips

It may take you awhile to fully master *SpeedScript*, but as you do you'll discover many ways to use the editing and formatting commands. For example, there is a simple way to simulate tab stops, say for a columnar table. Just type a period at every tab stop position. Erase the line with CTRL-E, then restore it with CTRL-*carat* multiple times. When you are filling in the table, just use word-left/word-right to jump quickly between the periods. Or you can use the programmable printkeys to embed your printer's own commands for setting and jumping to tab stops.

You don't have to change or define printer commands every time you write. Just save these definitions, and load this file for each session. You can create many custom definition files and have them ready to use on disk. You can create customized "fill-in-the-blank" letters. Just type the letter, and everywhere you'll need to insert something, substitute a unique character, such as an * or a CTRL character. When you're ready to customize the letter, use Find to locate each symbol and insert the specific information. Instead of typing an oft-used word or phrase, substitute a unique character, then use CTRL-Y to automatically change these characters into the actual word or phrase. You can even use *SpeedScript* as a simple filing program. Just type in all your data, flagging each field with a unique character. You can use Find to quickly locate any field.

If you experience any problems with *SpeedScript* that you are sure are not due to your error, please *write* (don't call) with a detailed explanation of the problem and how it occurred. Describe your hardware configuration. It also helps to send us a disk copy of your typing so we can determine with our equipment whether you have a hardware problem.

Due to the volume of mail, we cannot always reply to individual questions, but we welcome your suggestions. Who knows—your feedback may help make *SpeedScript 4.0* a reality.

Program 1: *SpeedScript 3.0* For Apple

Please refer to the "Apple MLX" article before entering this listing

```
START ADDRESS:  0800
END ADDRESS:    1E45

0800: 20 58 FC A9 26 A0 1E 85 E5
0808: ED 84 EE A0 00 20 DE 09 5F
0810: 20 E9 09 C9 D9 F0 0C C9 DC
0818: CE D0 F5 A9 00 8D 00 03 56
0820: 4C 28 08 A9 80 8D 00 03 39
0828: 20 29 0A 20 98 09 4C 18 87
0830: 08 A5 06 8D 51 08 A5 08 C6
0838: 8D 52 08 A5 19 8D 54 08 AE
0840: A5 18 8D 55 08 A6 F9 F0 80
0848: 20 A9 00 8D 53 1E A0 00 FF
0850: 89 FF FF 99 FF FF C8 CC 35
0858: 53 1E D0 F4 EE 52 08 EE C2
0860: 55 08 E0 00 F0 07 CA D0 43
0868: E0 A5 1D D0 DE 60 A5 F9 C0
0870: AA 05 1D D0 01 60 18 8A 0C
0878: 65 08 8D 99 08 A5 06 8D F8
0880: 98 08 18 8A 65 18 8D 9C D9
0888: 08 A5 19 8D 98 08 E8 A4 75
```

```
0B90: 1D D0 04 F0 0D A0 FF B9 97
089B: FF FF 99 FF FF BB C0 FF 7F
0BA0: D0 F5 CE 99 0B CE 9C 0B C6
0BAB: CA D0 EA 60 A2 00 BD 6D B3
0BB0: 09 BD 0F 09 BD 40 09 BD 5B
0BBB: B4 09 BD 0E 09 BD 3F 09 E5
0BC0: AD 4F 1E BD DC 0B BD ED 2B
0BCB: 0B BD 03 09 AD 50 1E BD A9
0BD0: DD 0B BD EE 0B BD 04 09 F9
0BDB: EB A0 00 B9 FF FF CB C9 7C
08E0: B2 F0 1C C9 3C F0 1B C0 25
0BEB: 28 D0 F0 BB B9 FF FF C9 7F
0BF0: B0 F0 0B 29 3F C9 20 F0 C3
0BFB: 05 BB D0 F0 A0 27 CB B4 BE
0900: ED B8 B9 FF FF 10 06 C9 7B
090B: 83 80 02 A5 FF 99 FF FF 09
0910: B8 10 EF A4 ED 18 98 6D 21
0918: DC 08 BD DC 08 8D ED 08 74
0920: 8D 03 09 AD DD 08 69 00 97
0928: 8D DD 08 8D EE 08 8D 04 09
0930: 09 E0 01 D0 03 8C 4E 1E 31
0938: C0 28 F0 0A A9 A0 99 FF 76
0940: FF C8 C0 28 D0 F8 8D 6D 72
094B: 09 8D 0F 09 8D 40 09 8D F1
0950: B4 09 8D 0E 09 BD 3F 09 7F
0958: E8 E0 18 F0 03 4C D9 08 2E
0960: AD DC 08 BD 59 1E AD DD D6
0968: 08 8D 5A 1E 60 04 05 05 31
0970: 06 06 07 07 04 04 05 05 97
097B: 06 06 07 07 04 04 05 05 9F
0980: 06 06 07 07 80 00 80 00 6D
0988: B0 00 B0 28 A8 28 A8 28 CC
0990: AB 2B AB 50 D0 50 D0 50 D4
0998: D0 50 D0 AD 46 1E 85 EB BD
09A0: 8D 4F 1E 8D 55 1E B5 FB 14
09AB: AD 47 1E 85 EC BD 50 1E DB
09B0: BD 56 1E 85 FC 38 AD 49 A8
09B8: 1E ED 47 1E AA A9 A0 A0 FD
09C0: FF C6 EC 91 EB C8 E6 EC 78
09C8: 91 E8 C8 D0 F8 E6 EC CA E4
09D0: D0 F6 91 E8 60 85 ED 84 73
09DB: EE 20 B0 FE A0 00 B1 ED C0
09E0: F0 06 20 ED FD CB D0 F6 78
09E8: 60 20 F5 09 F0 F8 C9 C0 4E
09F0: 90 02 29 DF 60 2C 00 C0 63
09F8: 10 2C AD 00 C0 C9 C0 90 13
0A00: 21 2C 00 03 30 1C 85 E3 C0
0A08: AD 61 C0 0D 44 1E 4D 63 CC
0A10: C0 10 08 A5 E3 09 20 8D F4
0A18: 10 C0 60 A9 00 8D 44 1E E7
0A20: A5 E3 8D 10 C0 60 A9 00 BD
0A2B: 60 20 58 FC A9 00 8D 52 0A
0A30: 1E 8D 46 1E 8D 4B 1E BD BB
0A38: 4A 1E BD 4C 1E BD F0 1E 96
0A40: 8D 19 1F 20 95 FE A9 60 A3
0A4B: 8D D5 A6 A9 01 20 5D A2 ED
0A50: A9 22 1B 69 01 BD 47 1E 46
0A59: A9 90 8D 49 1E A9 91 8D F3
0A60: 48 1E A9 99 8D 4D 1E A9 F7
0A68: FF 8D EE 1E 20 7B 0E A9 48
0A70: A0 BD F2 03 A9 0A 8D F3 4B
0A78: 03 49 A5 8D F4 03 A9 DF D4
0A80: B5 FF 20 90 0A A9 D1 A0 9F
0A88: 1C 20 D5 09 EE 51 1E 60 57
0A90: 20 A6 0A A9 C1 A0 1C 20 23
0A98: D5 09 A9 00 BD 51 1E 60 5D
0AA0: 20 B2 0A 4C 1B 0B A2 27 C4
0AAB: A9 20 9D 00 04 CA 10 FA B3
0AB0: A9 00 85 24 B5 25 4C 24 0A
0A8B: FC 2C 00 03 30 0D C9 C0 90
0AC0: 90 09 C9 E0 90 03 29 DF 69
0ACB: 60 29 1F 60 A0 00 81 F8 A5
0AD0: 85 1F A0 00 A5 1F C9 3C FC
0AD8: F0 08 29 3F C9 20 D0 0A BB
0AE0: A9 B0 4C EC 0A A9 B2 4C 8A
0AEB: EC 0A A9 81 91 FB 20 AC AC
0AF0: 0B A0 00 A5 1F 91 FB A2 65
0AFB: 12 20 F5 09 D0 19 C8 D0 BA
0800: F8 CA D0 F5 20 AC 08 A2 25
0B08: 12 20 F5 09 D0 09 CB D0 88
0810: F8 CA D0 F5 4C D2 0A 60 F0
0818: AD 45 1E 30 12 D0 0B A9 80
0820: 00 8D 44 1E CE 45 1E 4C 1B
0828: 2F 08 A9 00 BD 45 1E 20 AB
0B30: CC 0A AA AD 51 1E F0 07 4B
0B3B: BA 4B 20 90 0A 6B AA BA B4
0B40: C9 FF D0 06 20 5A 0F 4C BA
0B4B: 1B 0B BA C9 8D D0 02 A2 71
0B50: 3C BA 29 7F C9 20 90 4B 7C
0B5B: 8A 20 B9 0A 4B A0 00 B1 0A
0B60: FB C9 3C F0 05 AD 52 1E 1F
0B6B: F0 03 20 BE 10 6B A0 00 0B
0B70: 91 FB 20 AC 0B 3B A5 FB B5
0B7B: ED 55 1E B5 ED A5 FC ED E4
0BB0: 56 1E 05 ED 90 0E A5 FB CC
0BBB: 69 00 BD 55 1E A5 FC 69 45
0B90: 00 BD 56 1E E6 FB D0 02 B1
0B98: E6 FC 20 1F 0C 4C 1B 0B 24
0BA0: BA 4B 20 AC 0B 6B AE C7 E3
0BAB: 0B DD C7 0B F0 06 CA D0 6B
0BB0: F8 4C 18 08 CA 8A 0A AA 49
0888: A9 08 4B A9 17 48 8D E6 46
08C0: 08 48 8D E5 08 48 60 1D D8
0BC8: 95 88 8B 8A 82 8F 84 80 C1
0BD0: B9 85 8C 93 83 9E 9B 9A 3A
0BD8: 90 96 B7 97 BE B6 81 94 6D
08E0: 91 92 9D 99 98 D4 0C E8 E2
08E8: 0C 94 0D 08 0E 59 0F 8D FD
08F0: 10 83 0F C8 0F 23 11 E4 6A
08F8: 11 E3 13 EE 12 C8 14 58 22
0C00: 15 EC 15 78 0D 82 16 8C 63
0C0B: 13 74 0F FC 15 65 0E 85 B8
0C10: 1A 98 1C 7A 10 64 10 5C 15
0C18: 18 A2 10 94 1A 82 1C 20 A5
0C20: 7D 0C 38 A5 F8 ED 4F 1E AF
0C28: A5 FC ED 50 1E 80 20 38 41
0C30: AD 4F 1E ED 46 1E 85 ED 39
0C3B: AD 50 1E ED 47 1E 05 ED B8
0C40: F0 0D A5 F8 8D 4F 1E A5 14
0C4B: FC 8D 50 1E 20 AC 08 38 2A
0C50: AD 59 1E E5 FB 85 EB AD 33
0C58: 5A 1E E5 FC 85 EC 05 EB B7
0C60: F0 02 B0 18 18 AD 4F 1E 3D
0C68: 6D 4E 1E 8D 4F 1E AD 50 06
0C70: 1E 69 00 8D 50 1E 20 AC 82
0C78: 08 4C 4F 0C 60 38 AD 55 E6
0CB0: 1E ED 48 1E 85 ED AD 56 A3
0CB8: 1E ED 49 1E 05 ED 90 0C 43
0C90: AD 48 1E 8D 55 1E AD 49 F5
0C98: 1E 8D 56 1E 38 A5 F8 ED 0E
0CA0: 46 1E B5 ED A5 FC ED 47 37
0CA8: 1E 05 ED 80 08 AD 46 1E 93
0C80: 85 FB AD 47 1E 85 FC 60 16
0C88: 38 A5 F8 ED 55 1E 85 ED D0
0CC0: A5 FC ED 56 1E 05 ED 80 9F
0CCB: 01 60 AD 55 1E 85 F8 AD 31
0CD0: 56 1E 85 FC 60 AD 61 C0 59
0CD8: 0D 44 1E 4D 63 C0 10 55 84
0CE0: E6 FB D0 02 E6 FC 4C 1F 88
0CEB: 0C AD 61 C0 0D 44 1E 4D AD
0CF0: 63 C0 10 08 A5 FB D0 02 5E
0CF8: C6 FC C6 F8 4C 1F 0C A5 EB
0D00: F8 85 E8 A5 FC 85 EC C6 EF
0D0B: EC A0 FF B1 EB C9 A0 F0 94
0D10: 04 C9 3C D0 03 BB D0 F3 03
0D18: 81 E8 C9 A0 F0 0B C9 3C C0
0D20: F0 04 88 D0 F3 60 38 98 FB
0D28: 65 EB 85 F8 A5 EC 69 00 14
0D30: 85 FC 4C 1F 0C A0 00 B1 5C
0D38: F8 C9 A0 F0 08 C9 3C F0 B6
0D40: 04 CB D0 F3 60 CB D0 08 BA
0D4B: E6 FC A5 FC CD 56 1E 90 2E
0D50: 02 D0 19 81 F8 C9 A0 F0 17
0D58: EC C9 3C F0 E8 1B 98 65 30
0D60: FB 85 F8 A5 FC 69 00 B5 C6
0D6B: FC 4C 1F 0C AD 55 1E B5 3D
0D70: F8 AD 56 1E B5 FC 4C 1F 7B
0D78: 0C A9 00 8D 4F 1E AD 56 B0
0D80: 1E 38 E9 04 CD 47 1E B0 AD
0DBB: 03 AD 47 1E 8D 50 1E 20 64
0D90: AC 0B 4C 6C 0D AD 61 C0 F5
0D98: 0D 44 1E 4D 63 C0 30 03 64
0DA0: 4C 8A 11 A5 F8 85 EB A5 73
0DA8: FC B5 EC C6 EC A0 FF 81 4B
0DB0: E8 C9 AE F0 0C C9 A1 F0 D3
0D88: 0B C9 8F F0 04 C9 3C D0 E0
0DC0: 04 88 D0 EB 60 81 EB C9 43
0DC8: AE F0 1B C9 A1 F0 17 C9 3F
0DD0: BF F0 13 C9 3C F0 0F 8B 52
0DD8: D0 EB C6 EC A5 EC CD 46 C0
0DE0: 1E B0 E2 4C FC 0D B4 ED 6A
0DEB: C6 ED C8 F0 0A B1 EB C9 C2
0DF0: A0 F0 F7 BB 4C 26 0D A4 DB
0DF8: ED 4C C5 0D AD 46 1E B5 EE
0E00: FB AD 47 1E B5 FC 4C 1F 28
0E0B: 0C AD 61 C0 0D 44 1E 4D D0
0E10: 63 C0 30 03 4C 69 11 A0 0F
0E1B: 00 B1 FB C9 AE F0 1D C9 F9
0E20: A1 F0 19 C9 BF F0 15 C9 BE
0E2B: 3C F0 11 CB D0 EB E6 FC 4E
0E30: A5 FC CD 56 1E F0 E2 90 BB
0E3B: E0 4C 6C 0D CB D0 0E E6 C2
0E40: FC A5 FC CD 56 1E 90 05 12
0E4B: F0 03 4C 6C 0D B1 FB C9 DE
0E50: A0 F0 E9 C9 AE F0 E5 C9 A1
0E58: A1 F0 E1 C9 8F F0 DD C9 A1
0E60: 3C F0 D9 4C 5D 0D 20 A6 DC
0E68: 0A A9 38 A0 1E 20 D5 09 28
0E70: 20 38 11 F0 03 4C 90 0A 50
0E78: AD 4A 1E 8D CA 1E AD 48 10
0E80: 1E 8D C8 1E 20 A6 0A A9 C3
0E88: F0 A0 1C 20 D5 09 A9 01 F1
0E90: BD 51 1E 60 38 A5 F8 ED CF
0E98: 46 1E 85 ED A5 FC ED 47 33
0EA0: 1E 05 ED D0 03 68 68 60 C2
0EA8: A5 F8 85 06 A5 FC 85 08 D8
0E80: 60 38 A5 FB 85 19 49 FF A2
0E88: 65 06 8D CE 1E A5 FC 85 AE
0EC0: 1B 49 FF 65 0B 8D CF 1E 47
0EC8: A5 06 8D D0 1E A5 08 8D 1D
0ED0: D1 1E A5 19 8D D2 1E 85 1D
0ED8: 06 A5 18 8D D3 1E 85 08 C7
0EE0: 38 AD CF 1E 6D C8 1E CD 05
0EE8: 4D 1E 90 10 20 A6 0A A9 9F
0EF0: FE A0 1C 20 D5 09 A9 01 61
0EFB: 8D 51 1E 60 AD CA 1E 85 54
0F00: 19 AD C8 1E 85 18 AD CE 34
0F08: 1E 85 1D 18 6D CA 1E 8D 1C
0F10: CA 1E AD CF 1E 85 F9 6D 36
0F1B: CB 1E 8D CB 1E 20 31 0B ED
0F20: AD D0 1E B5 06 AD D1 1E 0E
0F2B: B5 0B AD D2 1E 85 19 AD D4
0F30: D3 1E 85 1B 38 AD 55 1E 63
0F3B: E5 19 B5 1D ED 56 1E E5 FA
0F40: 1B B5 F9 20 31 0B 38 AD 56
0F4B: 55 1E ED CE 1E BD 55 1E 33
0F50: AD 56 1E ED CF 1E BD 56 E5
0F5B: 1E 60 20 94 0E 20 F4 0C D1
0F60: 20 81 0E 3B AD CA 1E E9 FE
0F6B: 01 BD CA 1E AD CB 1E E9 68
0F70: 00 8D C8 1E 60 20 E0 0C 9E
0F7B: 20 94 0E 20 F4 0C 20 B1 59
0F80: 0E 4C 63 0F 20 78 0E 20 35
0F88: A6 0A A9 0A A0 1D 20 D5 E1
0F90: 09 20 CC 0A C9 C0 90 02 E9
0F9B: 29 DF 4B 20 90 0A 6B 29 F4
0FA0: 7F C9 57 D0 09 20 94 0E EB
0FAB: 20 FF 0C 4C B1 0E C9 53 C9
0FB0: D0 09 20 94 0E 20 A3 0D 0C
0FBB: 4C B1 0E C9 50 D0 09 20 BF
0FC0: 94 0E 20 8A 11 4C B1 0E 84
0FCB: 60 38 A5 FB ED 4F 1E B5 0B
0FD0: ED A5 FC ED 50 1E 05 ED C0
0FDB: F0 08 AD 4F 1E B5 FB AD 89
0FE0: 50 1E B5 FC 60 AD 46 1E 93
0FEB: B5 FB AD 47 1E 85 FC 4C 40
0FF0: 1F 0C A5 F8 B5 EB B5 19 16
0FF8: A5 FC B5 EC 85 1B A0 00 B2
1000: B1 EB C9 A0 D0 1E C8 D0 98
100B: F7 A5 EC CD 56 1E 90 0F 63
1010: AD 55 1E B5 EB AD 56 1E 59
101B: B5 EC A0 00 4C 24 10 E6 44
1020: EC 4C 00 10 18 9B 65 EB A4
1028: B5 06 A9 00 65 EC B5 0B B3
1030: 3B AD 55 1E E5 19 B5 1D 20
103B: AD 56 1E E5 1B 85 F9 3B 02
1040: A5 06 E5 19 8D CE 1E A5 8C
104B: 0B E5 1B BD CF 1E 20 31 8A
1050: 0B 3B AD 55 1E ED CE 1E F1
105B: BD 55 1E AD 56 1E ED CF 0A
1060: 1E 8D 56 1E 60 AD 61 C0 DC
106B: 0D 44 1E 4D 63 C0 30 03 3A
```

```
1070: 4C F2 0F A9 FF BD EB 1E 16
107B: 4C BD 10 A9 05 8D E8 1E 0D
10B0: 20 BD 10 B1 FB C9 A0 D0 4A
1088: 01 CB 4C 5D 0D A9 00 BD 57
1090: E9 1E 20 D4 10 A9 A0 AE 95
1098: E8 1E A0 00 91 FB CB CA A1
10A0: D0 FA 60 20 BE 10 20 8E 2B
10A8: 10 A9 3C A0 00 91 FB C8 D3
1080: 91 FB 20 AC 08 20 E0 0C F5
1088: 20 E0 0C 4C 78 10 A9 01 D7
10C0: BD EB 1E A9 00 BD E9 1E 68
10C8: 20 D4 10 A9 A0 A0 00 91 E3
10D0: F8 4C 1F 0C 18 AD 55 1E E6
10D8: 6D E8 1E AD 56 1E 6D E9 78
10E0: 1E CD 49 1E 90 05 68 68 60
10EB: 4C 23 11 18 A5 F8 85 06 C9
10F0: 6D E8 1E 85 19 A5 FC 85 FC
10FB: 08 6D E9 1E B5 1B 38 AD 4E
1100: 55 1E E5 06 85 1D AD 56 C3
1108: 1E E5 08 85 F9 20 6E 08 41
1110: 18 AD 55 1E 6D E8 1E 8D 0F
111B: 55 1E AD 56 1E 6D E9 1E 20
1120: 8D 56 1E 60 AD 52 1E 49 A4
1128: 0E 8D 52 1E F0 05 A9 1F EE
1130: 85 FF 60 A9 DF 85 FF 60 31
113B: A9 19 A0 1D 20 D5 09 20 E5
1140: CC 0A C9 C0 90 02 29 DF 4F
1148: C9 D9 60 20 A6 0A A9 30 84
1150: A0 1D 20 D5 09 20 38 11 85
1158: F0 03 4C 90 0A A2 FA 9A 81
1160: 20 98 09 20 82 0A 4C 18 89
1168: 08 A0 00 81 F8 C9 3C F0 C3
1170: 11 C8 D0 F7 E6 FC A5 FC 5A
1178: CD 56 1E 90 EE F0 EC 4C 45
1180: 6C 0D C8 D0 02 E6 FC 4C 34
11BB: 5D 0D A5 FB B5 EB A5 FC 35
1190: 85 EC C6 EC A0 FF 81 E8 AC
1198: C9 3C F0 11 88 C0 FF D0 F5
11A0: F5 C6 EC A5 EC CD 47 1E 82
11A8: 80 EC 4C FC 0D 38 98 65 97
1180: E8 85 E8 A9 00 65 EC 85 37
1188: EC 3B A5 EB E5 FB 85 ED EA
11C0: A5 EC E5 FC 05 ED D0 12 11
11C8: 84 ED 18 A5 E8 E5 ED 85 5E
11D0: E8 A5 EC E9 00 85 EC 4C CA
11D8: 9C 11 A5 E8 85 F8 A5 EC 55
11E0: 85 FC 4C 1F 0C AD 61 C0 18
11E8: 0D 44 1E 4D 63 C0 10 03 7C
11F0: 20 78 0E 20 A6 0A A9 3F F4
11F8: A0 1D 20 D5 09 20 CC 0A 80
1200: C9 C0 90 02 29 DF C9 D7 9F
1208: D0 09 20 3A 12 20 35 0D 07
1210: 4C 49 12 C9 D3 D0 09 20 9F
121B: 3A 12 20 17 0E 4C 49 12 99
1220: C9 D0 D0 09 20 3A 12 20 36
122B: 69 11 4C 49 12 C9 C1 D0 6F
1230: 03 4C 4B 11 20 1F 0C 4C 45
1238: 90 0A A5 F8 85 19 8D C4 0C
1240: 1E A5 FC 85 18 8D C5 1E 8D
1248: 60 38 A5 F8 85 06 ED C4 04
1250: 1E 8D CE 1E A5 FC B5 08 D6
125B: ED C5 1E 8D CF 1E 20 CB 81
1260: 0E AD C4 1E 85 F8 AD C5 AE
1268: 1E 85 FC 20 AC 08 4C FD 8A
1270: 11 A9 27 E5 24 8D 57 1E EE
127B: 20 84 FE A0 00 A9 1F 20 8C
1280: ED FD 8C 58 1E 20 F5 09 98
128B: F0 F8 AC 58 1E 85 ED A9 C8
1290: 88 20 ED FD A9 20 20 ED 9A
129B: FD A9 8B 20 ED FD A5 ED D9
12A0: C9 98 F0 37 C9 8D F0 39 C1
12AB: C9 FF F0 04 C9 88 D0 0F 31
1280: 88 10 04 C8 4C 7D 12 A9 50
1288: 88 20 ED FD 4C 7D 12 A5 E8
12C0: ED 29 7F C9 20 90 86 CC 30
12CB: 57 1E F0 81 A5 ED 20 89 38
12D0: 0A 99 B3 1E 20 ED FD C8 30
12DB: 4C 7D 12 20 B3 1C 4C 7D EA
12E0: 12 20 ED FD A9 00 99 83 87
12EB: 1E 98 20 80 FE A8 60 20 C9
12F0: A6 0A A9 76 A0 1D 20 D5 17
12FB: 09 20 4C 13 AD 46 1E 8D 84
1300: 72 AA AD 47 1E 8D 73 AA EC
130B: AD 55 1E 38 ED 46 1E 8D F3
1310: 6C AA AD 56 1E ED 47 1E B7
131B: BD 6D AA A9 30 BD 5F AA 71
1320: A9 09 BD 65 AA 20 60 14 10
132B: 20 B0 A1 20 96 14 AD C5 DA
1330: B5 D0 03 4C 53 14 20 A6 5C
133B: 0A AE C5 B5 20 02 A7 A9 25
1340: 87 20 F0 FD 60 A9 01 8D 69
1348: 51 1E 60 00 20 71 12 D0 66
1350: 06 20 90 0A 68 68 60 20 F9
135B: 95 A0 A9 00 8D 51 AA 8D 3B
1360: 52 AA 8D 74 AA 8D 66 AA 56
1368: 8D 6C AA 8D 6D AA 8D 63 33
1370: AA 8D 70 AA 8D 71 AA 8D 1D
1378: D3 9C A2 00 A0 01 89 B3 04
1380: 1E C9 8A D0 1F AD 83 1E 61
1388: 38 E9 B0 F0 0D 30 08 C9 73
1390: 03 80 07 8D 68 AA C8 4C E9
1398: A6 13 6B 68 20 A6 0A A2 8C
13A0: 02 4C 3C 13 A0 00 89 83 8F
13AB: 1E 20 2F 16 C9 E0 90 02 22
13B0: 29 DF 9D 75 AA E8 C8 CC C5
13BB: 58 1E D0 EA 60 20 A6 0A 36
13C0: A9 5F A0 1D 20 D5 09 20 04
13CB: CC 0A C9 C0 90 02 29 DF D8
13D0: 29 3F 09 40 48 AD 52 1E 3C
13DB: F0 03 20 8E 10 20 90 0A 54
13E0: 68 4C 6E 08 20 A6 0A A5 22
13EB: F8 CD 46 1E D0 07 A5 FC 16
13F0: CD 47 1E F0 05 A9 6A 8D D3
13FB: 27 04 A9 86 A0 1D 20 D5 E0
1400: 09 20 4C 13 AD 27 04 C9 4B
140B: 6A F0 03 20 9B 09 A5 F8 4C
1410: 8D 72 AA A5 FC 8D 73 AA FA
141B: A9 32 BD 5F AA A9 01 8D D4
1420: 65 AA 20 60 14 20 80 A1 73
142B: AE 60 AA AC 61 AA AD C5 86
1430: B5 F0 06 20 96 14 4C 36 06
143B: 13 8E 55 1E 8C 56 1E A5 B9
1440: F8 18 6D 55 1E BD 55 1E 5F
144B: A5 FC 6D 56 1E 8D 56 1E 87
1450: 20 96 14 20 A6 0A A9 7C DF
145B: A0 1D 20 D5 09 4C 45 13 90
1460: 2C 00 03 30 28 EE 56 1E E1
146B: A9 00 B5 08 AD 47 1E 85 E2
1470: 09 A0 00 81 0B C9 20 90 98
147B: 18 C9 C0 90 04 09 20 91 56
1480: 08 C8 D0 EF E6 09 A5 09 A7
148B: CD 56 1E D0 E6 CE 56 1E 38
1490: 60 09 C0 4C 7F 14 2C 00 AC
1498: 03 30 28 EE 56 1E A9 00 21
14A0: 85 08 AD 47 1E B5 09 A0 71
14AB: 00 81 08 C9 C0 90 08 C9 FC
14B0: E0 90 14 29 DF 91 08 C8 A0
14BB: D0 EF E6 09 A5 09 CD 56 F5
14C0: 1E D0 E6 CE 56 1E 60 29 08
14CB: 1F 4C 85 14 20 58 FC 20 08
14D0: 84 FE A9 00 8D D3 9C 20 45
14DB: 6E A5 A9 8D 20 ED FD A9 0E
14E0: 8C A0 1D 20 D5 09 20 F5 26
14EB: 09 C9 8D D0 F9 4C 90 0A F2
14F0: A2 00 8E C6 1E 8E C7 1E 81
14FB: 8E C8 1E 8E C9 1E 38 81 30
1500: E8 E9 80 90 2A C9 0A 80 F6
150B: 26 0E C6 1E 2E C7 1E 0E 5E
1510: C6 1E 2E C7 1E 0E C6 1E 3C
151B: 2E C7 1E 0E C6 1E 2E C7 C2
1520: 1E 0D C6 1E 8D C6 1E C8 E3
152B: D0 D4 E6 EC 4C FE 14 F8 18
1530: AD C6 1E 0D C7 1E F0 1C 2C
153B: 38 AD C6 1E E9 01 8D C6 D9
1540: 1E AD C7 1E E9 00 8D C7 F1
154B: 1E EE C8 1E D0 03 EE C9 72
1550: 1E 4C 30 15 AD C8 1E D8 99
155B: 60 38 AD CA 1E ED 4A 1E 7E
1560: 8D CC 1E AD C8 1E ED 48 21
156B: 1E 8D CD 1E 0D CC 1E D0 49
1570: 10 20 A6 0A A9 A1 A0 1D 52
157B: 20 D5 09 A9 01 8D 51 1E E2
1580: 60 18 A5 F8 85 06 6D CC 41
158B: 1E 85 19 A5 FC 85 08 6D 1C
1590: CD 1E 85 18 38 AD 55 1E CC
159B: E5 06 85 1D AD 56 1E E5 A2
15A0: 08 85 F9 18 65 18 CD 49 6D
15AB: 1E 90 10 20 A6 0A A9 99 54
15B0: A0 1D 20 D5 09 A9 01 8D 52
15BB: 51 1E 60 20 6E 08 18 AD 92
15C0: CC 1E 85 1D 6D 55 1E BD E5
15CB: 55 1E AD CD 1E B5 F9 6D 20
15D0: 56 1E 8D 56 1E A5 FB B5 C9
15DB: 19 A5 FC B5 1B AD 4A 1E 33
15E0: 85 06 AD 48 1E 85 08 20 F0
15EB: 31 08 4C 1F 0C A0 00 B1 BD
15F0: FB AA C8 81 F8 88 91 FB 19
15F8: C8 8A 91 F8 60 A0 00 BC 2E
1600: 44 1E 81 FB 2C 00 03 30 63
1608: 12 C9 20 90 09 C9 C0 90 3E
1610: 19 29 1F 4C 2A 16 09 C0 38
1618: 4C 2A 16 C9 C0 90 08 C9 7C
1620: E0 90 05 29 DF 4C 2A 16 AE
1628: 09 20 91 F8 4C E0 0C 2C FC
1630: 00 03 30 0A C9 20 90 07 8A
1638: C9 C0 90 02 09 20 60 09 3E
1640: C0 60 05 48 42 05 3A 01 D5
1648: 01 01 00 01 00 50 1B 0E CA
1650: 0F 12 8D EF 1E 8A 48 98 7D
1658: 48 38 AD DD 1E ED DF 1E D0
1660: AD DE 1E ED E0 1E 90 06 64
1668: AD EF 1E 20 ED FD 68 AB 0E
1670: 68 AA AD EF 1E 60 20 A6 89
1678: 0A A9 DC A0 1D 4C D5 09 88
1680: 4C EF 17 AD 61 C0 0D 44 FB
1688: 1E 4D 63 C0 10 03 4C D5 8A
1690: 16 20 A6 0A 20 AC 08 A9 82
169B: 84 A0 1D 20 D5 09 20 E9 E9
16A0: 09 A2 03 8E EA 1E C9 D3 7A
16AB: F0 30 A2 08 BE EA 1E C9 54
16B0: D0 D0 CD 20 A6 0A A9 CF B5
16B8: A0 1D 20 D5 09 20 E9 09 83
16C0: C9 81 90 0D C9 88 80 09 8C
16C8: 3B E9 B0 8D EA 1E 4C DA BD
16D0: 16 20 90 0A 60 A9 01 8D F8
16D8: EA 1E AD EA 1E A8 20 76 80
16E0: 16 AD EA 1E C9 03 D0 0D CB
16E8: AD 05 C3 18 6D 07 C3 C9 FF
16F0: 50 D0 05 A9 03 20 95 FE 77
16F8: 20 84 FE A2 00 8E D5 1E 64
1700: 8E D4 1E 8E E8 1E 8E EC 39
1708: 1E BD 42 16 9D D6 1E EB CB
1710: E0 0C D0 F5 A9 FF 8D E4 78
1718: 1E 8D E2 1E A2 04 8D 4D E4
1720: 16 9D 72 1F CA D0 F7 AD 3B
1728: 46 1E 85 E8 AD 47 1E 85 8C
1730: EC A0 00 8C E3 1E CC E2 D9
173B: 1E F0 06 AD D6 1E BD E3 78
1740: 1E 81 E8 30 07 C9 40 90 DA
1748: 03 4C CA 18 C9 3C F0 2C 33
1750: 99 C2 1F C8 EE E3 1E AD 5D
1758: E3 1E CD D7 1E 90 E2 8C 8C
1760: 54 1E 81 E8 C9 A0 F0 14 F8
1768: CE E3 1E 88 D0 F4 AC 54 48
1770: 1E 4C 7F 17 C8 81 E8 C9 D0
1778: A0 F0 01 88 8C 54 1E 98 66
1780: 3B 65 EB 85 E8 A5 EC 69 33
1788: 00 85 EC A0 00 AD E4 1E 5E
1790: C9 FF D0 03 20 75 18 AD A2
179B: E2 1E F0 03 20 A5 18 38 0E
17A0: 2E E2 1E AD 54 1E 8D 53 C6
17AB: 1E A9 C2 85 D6 A9 1F 85 22
17B0: D7 20 05 1C 20 86 18 AD EE
17B8: E4 1E CD DA 1E 90 03 20 A1
17C0: 00 18 38 A5 E8 ED 55 1E 36
17CB: 85 ED A5 EC ED 56 1E 05 C2
17D0: ED F0 2A 90 28 AD D5 1E 42
17DB: F0 08 A9 00 8D D4 1E 8D 01
17E0: D9 1E 20 00 18 AD EA 1E F2
17EB: C9 03 D0 03 20 E9 09 A9 6B
17F0: 00 20 95 FE A2 FA 9A 20 20
17FB: 90 0A 4C 18 08 4C 31 17 FF
1800: 38 AD D8 1E ED E4 1E A8 9C
180B: 88 8B F0 0A 30 08 A9 BD DF
1810: 20 52 16 88 D0 F8 AD D5 CB
1818: 1E F0 11 8D 53 1E A9 C2 87
1820: 85 D6 A9 21 85 D7 20 A5 81
1828: 18 20 05 1C A9 8D 20 52 E4
1830: 16 20 52 16 20 52 16 EE 84
1838: DD 1E D0 03 EE DE 1E AD 06
1840: DC 1E D0 31 AD EA 1E C9 82
184B: 03 F0 2A 38 AD DD 1E ED 0E
```

```
1850: DF 1E AD DE 1E ED E0 1E 24
1858: 90 18 A9 00 20 95 FE 20 42
1860: A6 0A A9 E8 A0 1D 20 D5 89
1868: 09 20 E9 09 20 76 16 AD A7
1870: EA 1E 20 95 FE AD D4 1E 71
1878: F0 11 8D 53 1E A9 C2 85 EE
1880: D6 A9 20 85 D7 20 A5 18 85
1888: 20 05 1C A9 8D 20 52 16 CF
1890: AC D9 1E 8C E4 1E 88 88 53
1898: F0 0A 30 08 A9 8D 20 52 60
18A0: 16 88 D0 F8 60 A9 A0 AC 3F
18A8: D6 1E 8C E3 1E F0 06 20 7C
18B0: 52 16 88 D0 FA 60 AC D8 3C
18B8: 1E 18 98 6D E4 1E 8D E4 87
18C0: 1E A9 8D 20 52 16 88 D0 EA
18C8: FA 60 8D E6 1E AE 10 19 93
18D0: DD 10 19 F0 09 CA D0 F8 34
18D8: CE E3 1E 4C 09 1A CA 8A C2
18E0: 0A AA 8C E5 1E A9 18 48 C0
18E8: A9 F3 48 8D 22 19 48 8D 93
18F0: 21 19 48 60 38 AD E5 1E 69
18F8: 65 E8 85 E8 A5 EC 69 00 F9
1900: 85 EC 4C 31 17 81 E8 C9 ED
1908: 3C F0 01 88 8C E5 1E 60 D5
1910: 10 57 4C 52 54 42 53 4E 6F
1918: 48 46 40 50 7F 58 4D 49 4E
1920: 47 7D 19 86 19 90 19 9A 88
1928: 19 A4 19 AE 19 88 19 C7 C3
1930: 19 E9 19 63 19 73 19 53 DE
1938: 19 49 19 40 19 02 1A 29 9E
1940: 1A C8 A9 00 8D E2 1E 4C 67
1948: 05 19 C8 20 F0 14 8D E1 33
1950: 1E 4C 05 19 C8 20 F0 14 93
1958: 8D DF 1E AD C9 1E 8D E0 AA
1960: 1E 4C 05 19 C8 20 F0 14 A3
1968: 8D DD 1E AD C9 1E 8D DE 38
1970: 1E 4C 05 19 C8 20 F0 14 83
1978: 8D D8 1E 4C 05 19 A9 00 10
1980: 8D DC 1E C8 4C 05 19 C8 72
1988: 20 F0 14 8D D6 1E 4C 05 2F
1990: 19 C8 20 F0 14 8D D7 1E 39
1998: 4C 05 19 C8 20 F0 14 8D 5C
19A0: D9 1E 4C 05 19 C8 20 F0 3E
19A8: 14 8D DA 1E 4C 05 19 C8 F6
19B0: 20 F0 14 8D D8 1E 4C 05 7F
19B8: 19 AC E5 1E C8 98 48 20 9A
19C0: 00 18 68 A8 8C E5 1E 60 29
19C8: 20 E2 19 88 8C D4 1E A0 04
19D0: 01 81 E8 99 C1 20 C8 CC F3
19D8: D4 1E 90 F5 F0 F3 C8 4C A3
19E0: 05 19 C8 81 E8 C9 3C D0 DF
19E8: F9 60 20 E2 19 88 8C D5 3C
19F0: 1E A0 01 81 E8 99 C1 21 FF
19F8: C8 CC D5 1E 90 F5 F0 F3 91
1A00: 4C 05 19 20 E2 19 4C 05 D9
1A08: 19 C8 81 E8 C9 8D F0 07 1E
1A10: 88 AD E6 1E 4C 4C 17 C8 3D
1A18: 20 F0 14 48 AD E6 1E AA 8F
1A20: 68 9D 42 1F 20 05 19 4C 8D
1A28: F4 18 A2 00 C8 81 E8 C9 DF
1A30: 3C F0 0A 20 2F 16 9D 83 92
1A38: 1E E8 4C 2C 1A 8E 58 1E D8
1A40: 20 98 09 20 57 13 A5 F8 DC
1A48: 8D 72 AA A5 FC 8D 73 AA 3F
1A50: A9 32 8D 5F AA A9 01 8D 19
1A58: 65 AA 20 60 14 20 80 A1 87
1A60: AE 60 AA AC 61 AA AD C5 FA
1A68: 85 F0 08 20 96 14 A9 00 6F
1A70: 20 95 FE 4C 36 13 8E 55 2F
1A78: 1E 8C 56 1E A5 F8 18 6D 46
1A80: 55 1E 8D 55 1E A5 FC 6D DC
1A88: 56 1E 8D 56 1E 20 96 14 39
1A90: 68 68 4C 27 17 20 C1 1A E5
1A98: AD F0 1E F0 16 20 68 18 CF
1AA0: 20 E7 1A AD EE 1E C9 FF 80
1AA8: F0 09 20 88 18 20 AC 08 0F
1AB0: 4C A0 1A 4C 90 0A AD 61 A4
1AB8: C0 0D 44 1E 4D 63 C0 30 A4
1AC0: 26 20 A6 0A A9 08 A0 1E 52
1AC8: 20 D5 09 20 71 12 8D F0 85
1AD0: 1E D0 03 4C 90 0A A0 00 58
1AD8: 89 83 1E 99 F1 1E C8 CC 8E
1AE0: 58 1E D0 F4 4C 90 0A A5 90
1AE8: F8 85 E8 A5 FC 85 EC A9 D5
1AF0: FF 8D EE 1E A0 01 A2 00 96
1AF8: AD F0 1E F0 50 8D F1 1E 8E
1B00: D1 E8 F0 05 E0 00 D0 EE 20
1B08: CA C8 D0 08 E6 EC A5 EC C3
1B10: CD 56 1E F0 02 80 36 E8 8D
1B18: EC F0 1E D0 E0 18 98 65 CF
1B20: E8 85 ED A5 EC 69 00 85 58
1B28: EE AD 55 1E C5 ED AD 56 65
1B30: 1E E5 EE 90 18 38 A5 ED 80
1B38: ED F0 1E 85 F8 8D ED 1E CD
1B40: A5 EE E9 00 85 FC 8D EE 6C
1B48: 1E 20 1F 0C 60 20 A6 0A 15
1B50: A9 0E A0 1E 20 D5 09 A9 E8
1B58: 01 8D 51 1E 60 AD 61 C0 88
1B60: 0D 44 1E 4D 63 C0 30 23 68
1B68: 20 A6 0A A9 18 A0 1E 20 D3
1B70: D5 09 20 71 12 8D 19 1F 07
1B78: F0 0E A0 00 89 83 1E 99 70
1B80: 1A 1F C8 CC 58 1E D0 F4 43
1B88: 4C 90 0A 38 A5 F8 85 19 0F
1B90: ED ED 1E 85 ED A5 FC 85 DA
1B98: 18 ED EE 1E 05 ED D0 62 78
1BA0: A9 FF 8D EE 1E 18 AD F0 E9
1BA8: 1E 65 F8 85 06 A9 00 65 58
1BB0: FC 85 08 38 AD 55 1E E5 30
1BB8: 19 85 1D AD 56 1E E5 18 6D
1BC0: 85 F9 20 31 08 38 AD 55 21
1BC8: 1E ED F0 1E 8D 55 1E AD 35
1BD0: 56 1E E9 00 8D 56 1E AD A6
1BD8: 19 1F F0 26 8D E8 1E A9 D9
1BE0: 00 8D E9 1E 20 D4 10 A0 AE
1BE8: 00 89 1A 1F 91 F8 C8 CC 9D
1BF0: 19 1F D0 F5 18 A5 F8 6D 81
1BF8: 19 1F 85 F8 A5 FC 69 00 E7
1C00: 85 FC 4C 1F 0C A0 00 CC 65
1C08: 53 1E F0 21 81 D6 30 04 EE
1C10: C9 40 80 1A 20 2F 16 20 FE
1C18: 52 16 AD EC 1E F0 0A A9 F5
1C20: 88 20 52 16 A9 DF 20 52 AF
1C28: 16 C8 4C 07 1C 60 8C E5 F8
1C30: 1E 8D E6 1E 20 2F 16 C9 4D
1C38: 43 D0 18 38 AD E1 1E ED 4C
1C40: 53 1E 4A 38 ED D6 1E A8 26
1C48: A9 A0 20 52 16 88 D0 FA 16
1C50: AC E5 1E 4C 29 1C C9 45 73
1C58: D0 11 38 AD D7 1E ED 53 85
1C60: 1E 38 ED D6 1E A8 A9 A0 68
1C68: 4C 4A 1C C9 55 D0 08 AD 25
1C70: EC 1E 49 01 8D EC 1E C9 06
1C78: 63 D0 12 8C E5 1E AE DD 84
1C80: 1E AD DE 1E 20 24 ED AC 08
1C88: E5 1E 4C 29 1C AE E6 1E DE
1C90: 8D 42 1F 20 52 16 4C 29 CA
1C98: 1C 20 A6 0A 38 AD 48 1E 83
1CA0: ED 55 1E AA AD 49 1E ED 50
1CA8: 56 1E 20 24 ED A9 01 8D 7F
1CB0: 51 1E 60 AD 44 1E 49 80 AD
1CB8: 8D 44 1E A9 01 8D 45 1E 0E
1CC0: 60 D3 D0 C5 C5 C4 D3 C3 41
1CC8: D2 C9 D0 D4 A0 83 AE 80 26
1CD0: 00 88 88 88 A0 C2 D9 EA
1CD8: A0 C3 AE A0 C2 D2 C1 CE E5
1CE0: CE CF CE A0 A6 A0 C8 AE 56
1CE8: A0 CD C1 D2 D4 C9 CE 00 85
1CF0: C2 D5 C6 C6 C5 D2 A0 CE CE
1CF8: D5 CC CC C5 C4 00 C2 D5 C6
1D00: C6 C6 C5 D2 A0 C6 D5 CC CD
1D08: CC 00 C4 C5 CC C5 D4 C5 8A
1D10: A0 A8 D3 AC D7 AC D0 A9 C6
1D18: 00 8A A0 C1 D2 C5 A0 D9 F9
1D20: CF D5 A0 D3 D5 D2 C5 8F 4E
1D28: A0 A8 D9 AF CE A9 8A 00 A5
1D30: C5 D2 C1 D3 C5 A0 C1 CC 78
1D38: CC A0 D4 C5 D8 D4 00 C5 D7
1D40: D2 C1 D3 C5 A0 A8 D3 AC 27
1D48: D7 AC D0 AC C1 A9 8A A0 49
1D50: D2 C5 D4 D5 D2 CE A0 D4 45
1D58: CF A0 C5 D8 C9 D4 00 C5 50
1D60: CE D4 C5 D2 A0 C6 CF D2 AF
1D68: CD C1 D4 A0 D6 C1 D2 C9 C8
1D70: C1 C2 CC C5 8A 00 D3 C1 71
1D78: D6 C5 8A 00 CE CF A0 C5 A3
1D80: D2 D2 CF D2 D3 00 CC CF 08
1D88: C1 C4 8A 00 D0 D2 C5 D3 5D
1D90: D3 A0 D2 C5 D4 D5 D2 CE 06
1D98: 00 CE CF A0 D2 CF CF CD CD
1DA0: 00 CE CF A0 D4 C5 D8 D4 D6
1DA8: A0 C9 CE A0 C2 D5 C6 C6 48
1DB0: C5 D2 AE 00 D0 D2 C9 CE 8C
1DB8: D4 A0 D4 CF 8A A0 D3 C3 E0
1DC0: D2 C5 C5 CE AC A0 D0 D2 D7
1DC8: C9 CE D4 C5 D2 8F 00 D3 F8
1DD0: CC CF D4 A0 CE D5 CD C2 36
1DD8: C5 D2 8F 00 D0 D2 C9 CE D6
1DE0: D4 C9 CE C7 AE AE AE 00 D8
1DE8: C9 CE D3 C5 D2 D4 A0 CE 8C
1DF0: C5 D8 D4 A0 D3 C8 C5 C5 F8
1DF8: D4 AC A0 D0 D2 C5 D3 D3 13
1E00: A0 D2 C5 D4 D5 D2 CE 00 DE
1E08: C6 C9 CE C4 8A 00 CE CF 83
1E10: D4 A0 C6 CF D5 CE C4 00 28
1E18: D2 C5 D0 CC C1 C3 C5 A0 5F
1E20: D7 C9 D4 C8 8A 00 CC CF 21
1E28: D7 C5 D2 C3 C1 D3 C5 A0 E1
1E30: D8 D9 AF CE DD 8F 60 00 62
1E38: CE D5 CC CC A0 C2 D5 C6 3A
1E40: C6 C5 D2 00 00 FF 00 00 A8
```

Program 2: *SpeedScript 3.0* File Converter

```
5 HIMEM: 8191
10 HOME
20 D$ = CHR$ (4)
30 PRINT D$;"MAXFILES1"
40 PRINT "DD YOU WANT TD:"
50 PRINT " (1) MAKE A SPEEDSCRI
   PT FILE INTO A TEXT      FILE
   "
60 PRINT " (2) MAKE A TEXT FILE
    INTO A SPEEDSCRIPT      FILE
   "
70 GET A$:A = VAL (A$)
80 IF A < > 1 AND A < > 2 THEN
   70
90 ON A GDTD 100,200
100 PRINT "ENTER SPEEDSCRIPT FI
    LE NAME": INPUT ":";A$
110 PRINT "ENTER TEXT FILE NAME
     TD CREATE": INPUT ":";8$
120 PRINT D$;"8LDAD ";A$;",A$20
    00"
125 L = PEEK (43616) + PEEK (43
    617) * 256 + 8192
130 PRINT D$;"OPEN ";8$
131 PRINT D$;"DELETE ";8$
132 PRINT D$;"OPEN ";8$
140 PRINT D$;"WRITE ";8$
150 FDR I = 8192 TD L - 1
155 8 = PEEK (I)
160 IF 8 = 60 THEN 8 = 141
170 PRINT CHR$ (8);
180 NEXT
190 PRINT
195 PRINT D$;"CLDSE ";8$
196 END
200 PRINT "ENTER TEXT FILE NAME
    ": INPUT ":";8$
210 INPUT "ENTER SPEEDSCRIPT FI
    LE NAME TD CREATE    :";A$
220 PRINT D$;"DPEN ";8$
230 PRINT D$;"READ ";8$
235 L = 0
240 DNERR GDTD 270
250 GET C$:A = ASC (C$ + CHR$ (
    0)) + 128: IF A = 141 THEN
    A = 60
260 PDKE 8192 + L,A:L = L + 1:
    GDTD 250
270 PDKE 216,0
280 PRINT
290 PRINT D$;"CLDSE ";8$
295 PRINT D$;"8SAVE ";A$;",A819
    2,L";L - 1
296 END
```

CAPUTE!

Modifications or Corrections To Previous Articles

Apple Games In ProDOS

Most of the Apple game programs recently published in COMPUTE! fail to operate properly with Apple's new ProDOS operating system, although they all work with DOS 3.3. The programs suffer from the same bug: Their graphics were developed using the DOS 3.3 version of the "Apple SuperFont" utility (published in the April 1985 issue). To use the following programs with ProDOS, these changes are required:

For "Mindbusters" (April 1985, p. 54):

```
120 POKE 6,0:POKE 7,141:P
    RINT CHR$(4);"PR# A$3
    00"
450 PRINT CHR$(4);"PR#0"
```

For "Space Caverns" (March 1985, p. 54):

```
910 HOME:HGR:POKE 6,0:POK
    E 7,141:PRINT CHR$(4)
    ;"PR# A$300"
```

For "Bowling Champ" (February 1985, p. 126):

```
130 HOME : POKE 230,32: CALL
    62450: HGR: POKE 6,0: POK
    E 7,141: PRINT CHR$(4);"P
    R# A$300"
830 X = 0: FOR I = 768 TO 853
    : READ A:X = X + A: POKE
    I,A: NEXT : IF X < > 7950
     THEN PRINT "ERROR IN DAT
    A STATEMENTS FOR ML AT 76
    8.": STOP
840 DATA  216,133,69,134,70,1
    32,71,166,7
```

For "Paratrooper" (January 1985, p. 72):

```
200 FOR I = 768 TO I + 85: RE
    AD A:X = X + A: POKE I,A:
     NEXT : IF X < > 23417 TH
    EN PRINT "ERROR IN DATA S
    TATEMENTS.": STOP
220 PRINT  CHR$ (4)"PR# A$300
    "
1130 DATA  216,133,69,134,70,
     132,71,166,7
```

For "Things in the Dark" (December 1984, p. 79):

```
770 HOME : HGR : POKE 6,0: PO
    KE 7,141: PRINT CHR$ (4)"
    PR# A$300"
990 X = 0: FOR I = 768 TO 853
    : READ A:X = X + A: POKE
    I,A: NEXT : IF X < > 7950
     THEN PRINT "ERROR IN 1ST
     SET OF DATA STATEMENTS."
    : STOP
1000 DATA  216,133,69,134,70,
     132,71,166,7
```

For "Spiders" (November 1984, p. 96):

```
160 CALL 36884
```

In addition, ProDOS filename conventions require that you BSAVE the machine language portion of Spiders (Program 6) with the title SPIDER.2 instead of SPIDER 2 as indicated on p. 90. You'll also have to change SPIDER 2 to SPIDER.2 in line 130 of Program 5.

Atari Disk Rx

The author of this utility program from the March issue (p. 107) has provided the following corrections: Renumber line 4015 to become line 4006, replace the old line 4015 with 4015 REM, change the GOTO HALT in line 5075 with GOSUB HALT, and change the REC=1 in line 6040 to REC=0.

It is also possible for the program to crash with the message ERROR 5 IN LINE 325 while you are attempting to recover large files. The error occurs because the program does not check for buffer overflow. To prevent this, reader Jim Owens suggests adding and changing the following lines:

```
323 IF FILL=1 AND (8CNT+T
    YPE)>RAM THEN ?:? "BU
    FFER FULL. LAST BUFFE
    R SECT=";SECT-1:POP:G
    OTO 330
326 NEXT SECT
330 IF DIR=1 THEN RETURN
```

Apple SuperFont

There are two typos in the checksum program (Program 6, p. 137) for this custom character utility for the Apple II series in the April issue. Lines 140 and 150 should read as follows:

```
140 FOR J = 0 TO 63:S = S
     + PEEK (4096 + I * 6
    4 + J): NEXT
150 READ A:S = S - 256 *
    INT (S / 256)
```

Plus/Term For VIC & 64

In addition to the items in last month's "CAPUTE!" column, there is another correction to the machine language portion for the VIC-20. Whenever you load a file into the buffer, the lower boundary of the buffer is reset incorrectly so that 256 bytes of garbage are added to the start of the text. To remedy this, reset the VIC by turning it off and back on. Load the machine language data by typing LOAD "PLUS/TERM.ML",8,1 (for tape, replace the ,8,1 with ,1,1). Then enter the following line in direct mode (without a line number):

```
POKE 43,0: POKE 44,24: POKE 7075,64
```

Immediately save the revised version by typing SAVE "PLUS/TERM.ML2",8 (or ,1 for tape). Now change line 100 of the BASIC portion to reflect the new name for the machine language portion.

If your printer is not properly handling upper- and lowercase characters, try changing line 1900 to 1900 OPEN 5,ZE,7.

The article states that changing the baud rate after other parameters have been set causes all other parameters to revert to their default values. Actually, even though the various submenus will show all parameters reverted to their default states, the actual parameter values will not have changed. Thus, the menus will not reflect the actual settings of the parameters. For this reason, you should always change the baud rate before changing any other parameter. ©

Advertisers Index

Reader Service Number/Advertiser	Page
102 Abacus Software	79
103 Abacus Software	81
104 American PEOPLE/LINK	69
Apple Computer, Inc.	14–15
105 Apricot, Inc.	2–3
106 Aprotek	4
Batteries Included	13
107 CALABCO Peripherals Division	32–33
108 Cardco, Inc.	IBC
Commodore	BC
109 CompuServe	21
ComputAbility	47
Computel Publishing Society	105
110 Computer Air Filters	127
111 Computer Direct	62–63
112 Computer Mail Order	38–39
113 Comspec Communications Inc.	65
114 Covox Inc.	99
115 Davidson & Associates, Inc.	7
116 Davidson & Associates, Inc.	107
Disk World!, Inc.	111
117 EPYX	25
Family Discount Computer Products	109
118 Grolier Electronic Publishing, Inc.	31
119 Harmony Video & Computers	115
ICS Computer Training	83
120 Indus Systems	43
121 J & R Music World	87
Jason-Ranheim	80
Lyco Computer	100–101
Maxell	11
MICROpendium	105
122 Mimic Systems, Inc.	27
North Hills Corporation	99
North Hills Corporation	109
123 Nibble Notch Computer Products	67
124 Okidata	IFC–1
125 Pacific Exchanges	109
126 PDS Sports	29
127 Precision Data Products	109
128 Prof. Jones	107
129 Protecto	60–61
130 PSI	9
131 Quinsept, Inc.	16

Reader Service Number/Advertiser	Page
132 Starpoint Software	28
133 Strategic Simulations, Inc.	37
134 SubLOGIC Corporation	23
135 Tripp Lite	64
136 Ultima Electronics, Ltd.	86
137 Witt's End	80

COMPUTE! Books	44–45
COMPUTE! Books' New Releases	41
COMPUTE! Classifieds	59
COMPUTE! Subscription	17
COMPUTE!'s Apple Applications	48
COMPUTE!'s ML for Beginners and Second Book of ML	35

Technical

Breakthrough

Number 27

Price and performance

no other printer interface can touch!

Only CARDCO would dare improve on its own best seller (the +G has far out-sold any other printer interface, and has set the industry standard by which competitors are judged). The G-Wiz is even better — and costs 30% less.* Our 27th major innovation in Commodore accessories has all the +G's features, and more...

Built-In Buffer for More Speed
The G-Wiz buffer dumps high-resolution screens up to 18 times faster than competitive interfaces without buffers. No more waiting! A complex 50-minute printout now takes just 2.5 minutes with the G-Wiz.

Exclusive! Aspect Ratio Selection
Only the G-Wiz matches graphics characters to standard characters on Okidata, Epson, Star Micronics, Prowriter, Smith Corona, C-ITOH, Gorilla Banana, end many other dot matrix printers. Now you can perfectly align high resolution graphics characters within text blocks, or in columns.

CARDCO excellence triumphs again! The G-Wiz is the "best bang for the buck" on the printer interface market today — and it's backed by CARDCO'S exclusive lifetime warranty! G-Wiz: another distant target for the competition to shoot at.

CARDCO, Inc. 300 S. Topeka/Wichita, KS 67202

* Actual price may vary according to dealer and region

The Wizards from the Land of Oz Have Done It Again!

IT'S NOT HOW MUCH YOU PAY.

IT'S HOW MUCH YOU GET.

The computer at the top has a 64K memory.

It has the initials I, B, and M. And you pay for those initials.

The Commodore 64™ has a 64K memory.

But you don't pay for the initials, you just pay for the computer. About one third the price of the IBM PCjr.™

The Commodore 64 also has a typewriter-type keyboard with 66 typewriter-type keys. (Not rubber chicklet keys like the IBM PCjr.)

It has high resolution graphics with 320 x 200 pixel resolution, 16 available colors and eight 3-dimensional sprites.

It has 9-octave high fidelity sound.

The Commodore 64 is capable of running thousands of programs for home and office. And if you add a printer or color monitor, disk drive and a modem—all together it just about equals the price of the IBM PCjr all alone. With no peripherals.

So you can buy a computer for a lot of money.

Or buy a lot of computer for the money.

COMMODORE 64

IT'S NOT HOW LITTLE IT COSTS,
IT'S HOW MUCH YOU GET.

RETROMAGS

Our goal is to preserve classic video game magazines so that they are not lost permanently.

People interested in helping out in any capacity, please visit us at www.retromags.com.

No profit is made from these scans, nor do we offer anything available from the publishers themselves.

If you come across anyone selling releases from this site, please do not support them and do let us know.

Thank you!